

**UNIVERSIDADE METODISTA DE PIRACICABA**

**FACULDADE DE ENGENHARIA, ARQUITETURA E URBANISMO  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO**

**Método para Geração Automática da Trajetória da  
Ferramenta para a Usinagem de Rebarbas em Peças  
Fundidas: Uma Contribuição para a Automação do  
Processo de Rebarbação**

**Msc. Eng. Bruno Brogio de Oliveira**  
Orientador: Prof. Dr.-Ing. Klaus Schützer

Santa Bárbara d'Oeste, SP

2016

**UNIVERSIDADE METODISTA DE PIRACICABA**

**FACULDADE DE ENGENHARIA, ARQUITETURA E URBANISMO  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO**

**Método para Geração Automática da Trajetória da Ferramenta para a Usinagem de Rebarbas em Peças Fundidas: Uma Contribuição para a Automação do Processo de Rebarbação**

**Msc. Eng. Bruno Brogio de Oliveira**

Orientador: Prof. Dr.-Ing. Klaus Schützer

Tese de Doutorado apresentada no Programa de Pós-Graduação em Engenharia de Produção da Faculdade de Engenharia, Arquitetura e Urbanismo, da Universidade Metodista de Piracicaba – UNIMEP.

Santa Bárbara d'Oeste, SP

2016

## Agradecimentos

A minha família, em especial a minha mãe, Maristela por todo apoio, ajuda e paciência.

Ao Prof. Dr.-Ing. Klaus Schützer pela orientação e incentivo para a conclusão deste trabalho.

Ao CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico - "*National Counsel of Technological and Scientific Development*" pela bolsa de doutorado.

Ao SCPM, assim como seus integrantes. Em especial ao Carlos Miralles, o qual esteve presente nos ensaios contribuindo e participando.

A UNIMEP por ter me recebido como pesquisador.

Ao professor Reginaldo Teixeira Coelho, que além de contribuir como membro da banca, gentilmente cedeu o robô utilizado nesse experimento em seu laboratório, o NUMA, da USP São Carlos.

A RGB Sistemas em Santa Bárbara D'Oeste, a qual forneceu o escâner como equipamento assim como os softwares para sua utilização, além de disponibilizar ajuda e suporte para esse trabalho.

Aos meus amigos que sempre ouviram e apoiaram a conclusão desse trabalho.

# Sumário

Agradecimentos .....	III
Índice de Figuras.....	IV
Resumo.....	X
Abstract .....	XI
1 Introdução.....	1
2 Revisão Bibliográfica .....	6
2.1 Robôs industriais .....	6
2.1.1 Histórico .....	6
2.1.2 Características Quanto ao Uso da Robótica Industrial .....	9
2.1.3 Conceitos Básicos de um Robô Industrial .....	10
2.1.4 Classificação de robôs por quanto a estrutura mecânica .....	12
2.1.5 Sistemas de referência de um robô: .....	16
2.1.6 Programação de um Robô Industrial .....	18
2.1.7 Aplicabilidades de robôs industriais.....	21
2.2 Sistemas para a automação do processo de programação e controle de um robô industrial .....	27
2.2.1 Aplicações na substituição do trabalho manual complexo.....	27
2.2.2 Compensação de erros na usinagem com robôs .....	28
2.2.3 Geração de trajetória para robôs.....	32
2.2.4 Processos de acabamento utilizando robôs.....	33
2.3 Processo de escaneamento .....	38
2.3.1 Etapas para uso de escaners 3D na obtenção de nuvens de pontos a partir de modelos físicos. ....	42
2.3.2 Métodos de Calibração para escâner a laser 3D.....	43
2.4 Processos de escaneamento automatizados.....	45

2.5	Fundamentos Matemáticos .....	52
2.5.1	SpinImages .....	53
2.5.2	Algoritmo de Horn.....	56
2.5.3	ICP – Iterative Closest Point .....	59
2.5.4	Identificação de rebarbas.....	60
3	Objetivos e Metodologia.....	65
3.1	Objetivo.....	65
3.1.1	Objetivos Específicos .....	65
3.2	Metodologia.....	66
4	Apresentação geral do conceito e corpos de prova .....	68
4.1	Apresentação geral do conceito .....	68
4.2	Corpo de prova.....	74
4.2.1	Corpo de prova A.....	74
4.2.2	Corpo de Prova B.....	75
4.2.3	Projeto e Usinagem dos Corpos de prova.....	76
5	Implementação e Validação.....	78
5.1	Preparação do sistema.....	78
5.1.1	Fixação do Escâner .....	78
5.1.2	Fixação do Cabeçote Fresador .....	83
5.1.3	Mesa Rotativa.....	83
5.1.4	Zeramento do Robô .....	85
5.1.5	Zeramento da Ferramenta .....	85
5.1.6	Zeramento da Base. ....	89
5.2	Processo de Escaneamento .....	91

5.2.1	Fixação da peça .....	91
5.2.2	Escaneamento .....	93
5.2.3	Geração do Arquivo STL .....	95
5.3	Processo de Alinhamento .....	96
5.3.1	Aplicação do Método de Johnson .....	101
5.3.2	Aplicação do Algoritmo de Horn .....	106
5.3.3	Aplicação do ICP .....	108
5.4	Processo de Reconhecimento de Rebarbas.....	109
5.5	Processo de Geração da Trajetória da Ferramenta .....	110
5.5.1	Pós Processamento .....	113
5.5.2	Transferência dos arquivos para o comando do robô .....	114
5.6	Processo de Usinagem das Rebarbas.....	115
5.7	Análise dos Resultados.....	118
6	Conclusões .....	120
7	Referências.....	124
8	Anexos.....	132
8.1	Anexo A – Masterização do robô Kuka KR5 .....	132
8.2	Anexo B - Pós-Processamento Utilizando CamRob .....	143

## Índice de Figuras

Figura 1.1: Formação de rebarbas em peças usinadas [2] .....	1
Figura 1.2: Roda de alumínio obtida pelo processo de fundição .....	2
Figura 2.1: Robô Industrial de 6 graus de liberdade [16] .....	10
Figura 2.2: Associação ao braço humano [20] .....	11
Figura 2.3: Classificação de Robôs quanto a estrutura mecânica [19].....	12
Figura 2.4: Representação do robô tipo cartesiano onde: A – Representa a esquemática do robô cartesiano [20]. B- Representa o volume de trabalho do robô cartesiano [16]. .....	13
Figura 2.5: Representação do robô tipo cilíndrico onde: A – Representa a esquemática do robô cilíndrico [20]. B- Representa o volume de trabalho do robô cilíndrico [16].....	14
Figura 2.6: Representação do robô tipo esférico onde: A – Representa a esquemática do robô esférico [20]. B- Representa o volume de trabalho do robô esférico [16]. .....	14
Figura 2.7: Representação do robô tipo revolução onde: A – Representa a esquemática do robô revolução [20]. B- Representa o volume de trabalho do robô revolução [16]. .....	15
Figura 2.8: Representação do robô tipo scara onde: A – Representa a esquemática do robô scara [20]. B- Representa o volume de trabalho do robô scara [16]. .....	16
Figura 2.9: Definição dos ângulos de orientação RPY [20] .....	16
Figura 2.10: Referência Global [19].....	17
Figura 2.11: Referência Articulação [19] .....	17
Figura 2.12: Referência Ferramenta [19].....	18
Figura 2.13: Robô ABB IRB 6620 para manuseio 25 .....	22
Figura 2.14: Robô KUKA KR 6-2 destinado à operação de manuseio [26] .....	23
Figura 2.15: Robô ABB IRB 1600ID destinado a Soldagem [30].....	24
Figura 2.16: Robô IRB 52, destinado para pintura [32] .....	26
Figura 2.17: Robô para Usinagem ABB IRB 6660 [38].....	27

Figura 2.18: Robô e seu dispositivo para compensação de erros durante a usinagem [34] .....	29
Figura 2.19: Cabeçote compensativo proposto por Liao et al. [3].....	29
Figura 2.20: Fluxograma do processo do controle de torque pneumático [3].....	30
Figura 2.21: Proposta de Changhoon Kim et al [24].....	31
Figura 2.22: Esquema da malha de controle proposta pelo artigo. [40] .....	31
Figura 2.23: Proposta de Pan e Zhang para controle de forças na usinagem [8].....	32
Figura 2.24: Robô de arquitetura aberta usada pelo artigo de Nagata At All [41].....	33
Figura 2.25: Polimento [1].....	34
Figura 2.26: Polimento [1].....	34
Figura 2.27: Retificação [1].....	35
Figura 2.28: Rebarbação [1].....	35
Figura 2.29: Hardware do artigo [9].....	36
Figura 2.30: Arquitetura da metodologia do artigo [9].....	37
Figura 2.31: Geração de nuvem de pontos através de duas câmeras [45] .....	39
Figura 2.32: Câmeras montadas [45] .....	40
Figura 2.33: Fluxograma da aquisição de dados [47].....	42
Figura 2.34: Ilustração de como deve ser montado a câmera e a mesa rotativa. [44] .....	44
Figura 2.35: Ilustração do escaner portátil fixado ao robô [44] .....	44
Figura 2.36: Sugestão do trabalho para a calibração da mesa [44] .....	45
Figura 2.37: Dispositivo para escaneamento do objeto. [42].....	47
Figura 2.38: A esquerda, modelo do sistema. A direita, modelo físico do processo. [43] .....	47
Figura 2.39: Processo de cálculo de posições para o escaner [43].....	48
Figura 2.40: Processo físico escaneando objeto real [43] .....	49
Figura 2.41: Análise de uma lamina de turbina eólica [17] .....	50
Figura 2.42: Ilustração do método de Larsson e Kjellander para geração automática da trajetória do escaner através de sistemas CAD. [48] .....	51
Figura 2.43: Alinhamento de Imagens 3D [51] .....	53
Figura 2.44: SpinMaps e SpinImages [51].....	54
Figura 2.45: SpinMaps [51].....	55
Figura 2.46: SpinImages [51].....	55
Figura 2.47: Coordenadas dos pontos mensuradas em dois diferentes sistemas de coordenadas [52].....	57

Figura 2.48: ICP interatividade. [54] .....	60
Figura 2.49: Projeção de pontos modelo escaneado em uma faceta do modelo CAD. [52] .....	61
Figura 2.50: Motivo pelo qual o ponto projetado deve estar dentro do triângulo. [52] .	62
Figura 2.51: Ponto pertencente ao triângulo. [52].....	63
Figura 2.52: Ponto não pertencente ao triângulo. [52].....	64
Figura 3.1: Fluxograma da metodologia proposta .....	67
Figura 4.1: Ciclo geral do trabalho.....	68
Figura 4.2: Preparação do sistema.....	69
Figura 4.3: Ciclo do processo de escaneamento.....	71
Figura 4.4: Ciclo do processo de alinhamento.....	71
Figura 4.5: Ciclo do processo de separação de rebarbas .....	72
Figura 4.6: Ciclo do processo de geração de trajetórias .....	73
Figura 4.7: Modelo CAD do corpo de prova A .....	75
Figura 4.8: Modelo CAD do corpo de prova B .....	75
Figura 4.9: Projeto e manufatura do corpo de prova A .....	77
Figura 4.10: Projeto e manufatura do corpo de prova B.....	77
Figura 5.1: Hand Scan - ExaScan da Creaform .....	79
Figura 5.2: Go Scan. Equipamento usado para escanar a ExaScan.....	79
Figura 5.3: Arquivo STL sendo processado no Software GeoMagic .....	80
Figura 5.4: Detalhes do modelamento do dispositivo. ....	81
Figura 5.5: Imagens sequenciais do desenvolvimento e fabricação do dispositivo de fixação do escaner.....	82
Figura 5.6: As duas partes que abraçam o escaner como dispositivo de fixação e uma flange de montagem e distanciamento do robô. ....	82
Figura 5.7: Imagem do Spindle montado no seu suporte e por sua vez, montado na flange do robô.....	83
Figura 5.8: Fixação da mesa rotativa e da morsa.....	84
Figura 5.9: Sequencia de posições para a calibração do TCP da ferramenta.....	87
Figura 5.10: Sistema de coordenadas da ferramenta em relação ao global .....	88
Figura 5.11: Zeramento da base .....	90
Figura 5.12: Sequencia de posições para a calibração da base .....	90
Figura 5.13: Sistema de coordenadas da base em relação ao da ferramenta e ao global .....	91

Figura 5.14: Fixação dos corpos de prova na morsa.....	92
Figura 5.15: Peça A colocada no dispositivo de fixação.....	93
Figura 5.16: Peça B colocada no dispositivo de fixação.....	93
Figura 5.17: Robô guiando o escaner através do corpo de prova. ....	95
Figura 5.18: STL resultante do processo de escaneamento do corpo de prova A .....	96
Figura 5.19: STL resultante do processo de escaneamento do corpo de prova B .....	96
Figura 5.20: Corpo de prova A, modelado no sistema CAD NX e Salvo em formato STL.....	98
Figura 5.21: Corpo de prova B, modelado no sistema CAD NX e Salvo em formato STL.....	98
Figura 5.22: Etiquetas de referência do escaner .....	99
Figura 5.23: Comportamento do escaner e suas etiquetas .....	100
Figura 5.24: Arquivo CAD corpo de prova A salvo em STL aplicado a ferramenta de subdivisão de triângulos .....	102
Figura 5.25: Arquivo CAD corpo de prova B salvo em STL aplicado a ferramenta de subdivisão de triângulos .....	103
Figura 5.26: Antes e depois do comando de subdivisão das facetas ou triângulos...	103
Figura 5.27: Sequência de escolha de pontos das malhas .....	105
Figura 5.28: Distância dos pontos 2, em vermelho ao ponto 1, em azul .....	105
Figura 5.29: Sobreposição do arquivo CAD do corpo de prova A com o do arquivo escaneado .....	106
Figura 5.30: Sobreposição do arquivo CAD do corpo de prova B com o do arquivo escaneado .....	107
Figura 5.31: Corpo de Prova A alinhado .....	107
Figura 5.32: Corpo de Prova B alinhado .....	108
Figura 5.33: Rebarbas do corpo de prova A.....	109
Figura 5.34: Rebarbas do corpo de prova B.....	110
Figura 5.35: Região final após a rebarbação do corpo de prova A.....	111
Figura 5.36: Região final após a rebarbação do corpo de prova B.....	111
Figura 5.37: Geração da trajetória da ferramenta do corpo de prova A .....	112
Figura 5.38: Geração da trajetória da ferramenta do corpo de prova B .....	112
Figura 5.39: Possíveis posições para fixar a posição do Spindle .....	114
Figura 5.40: Corpo de prova A – Início da rebarbação .....	115
Figura 5.41: Corpo de prova B – Início da rebarbação .....	116

Figura 5.42: Corpo de prova A – Processo de rebarbação .....	116
Figura 5.43: Corpo de prova B – Processo de rebarbação .....	117
Figura 5.44: Corpo de prova A, com e sem rebarba.....	117
Figura 5.45: Corpo de prova B, com e sem rebarba.....	118
Figura 5.46: Corpo de Prova A, comparação dimensional. ....	119
Figura 5.47: Corpo de Prova B, comparação dimensional .....	119
Figura 8.1: Posição inicial de masterização.....	133
Figura 8.2: Modos de Operação do Robô. ....	133
Figura 8.3:Estado da movimentação dos eixos .....	134
Figura 8.4: Botões de Enable .....	135
Figura 8.5: Posicionamento do eixo 1 do robô. Figura A.....	135
Figura 8.6: Posicionamento do eixo 1 do robô. Figura B.....	136
Figura 8.7: Posicionamento do eixo 2 do robô. Figura A.....	136
Figura 8.8: Posicionamento do eixo 2 do robô. Figura B.....	137
Figura 8.9: Ilustração da localização do parafuso para masterizar o eixo 4. ....	137
Figura 8.10: Ilustração da localização do parafuso para masterizar o eixo 4, parafuso removido. ....	138
Figura 8.11: Jeito correto do cabo preto.....	138
Figura 8.12: Posicionamento do eixo 4 para a colocação do parafuso de masterização. ....	139
Figura 8.13: Posicionamento do eixo 4 após a colocação do parafuso de masterização. ....	139
Figura 8.14: Posicionamento do eixo 3. ....	140
Figura 8.15: Posicionamento do eixo 5. ....	140
Figura 8.16: Posicionamento do eixo 6 para a montagem da peça de masterização. ....	141
Figura 8.17: Montagem da peça de masterização.....	141
Figura 8.18: Posicionamento dos eixos para a masterização.....	142
Figura 8.19: Tela inicial do Kuka SimPro para início do pós processamento. ....	143
Figura 8.20: Novo trabalho .....	144
Figura 8.21: Add Process Comand.....	144
Figura 8.22: Parametros de velocidade em vazio.....	145
Figura 8.23: Erro no arquivo CLSF .....	145
Figura 8.24: Correção do arquivo CLSF .....	146
Figura 8.25: Comando Read finalizado. ....	147

Figura 8.26: Janela de seleção e configuração de base .....	147
Figura 8.27: Estado depois da inserção das coordenadas da base .....	148
Figura 8.28: Janela de filtro .....	148
Figura 8.29: Final da janela Filter .....	149
Figura 8.30: Janela de parametros de ferramenta.....	149
Figura 8.31: Parametros do zeramento da ferramenta.....	150
Figura 8.32: Finalização dos parametros da ferramenta .....	150
Figura 8.33: Janela de estratégias .....	151
Figura 8.34: Possiveis restrições de posicionamento do Spindle .....	151
Figura 8.35: Janela de finalização das estratégias.....	152
Figura 8.36: Estado da janela até a configuração da janela Strategy.....	152
Figura 8.37: Processamento do arquivo CLSF em ordem.....	153
Figura 8.38: Simulação de todos os movimentos .....	154
Figura 8.39: Geração dos arquivos para o robô .....	154

## Resumo

Em geral, os produtos passam por diversas e diferentes etapas durante sua manufatura. Uma dessas etapas é o acabamento. Na etapa de acabamento, pode ser citado o processo de rebarbação. Rebarbas são caracterizadas como projeções indesejáveis na peça ou produto e essas precisam ser removidas. São duas as principais razões que justificam sua remoção, aspectos visuais ou ainda, aspectos mecânicos. Rebarbas podem causar acúmulos de tensões e forças que podem levar a peça a fadiga ou ainda ruptura. Dessa forma, há a necessidade de processos para sua remoção. Atualmente esse processo é feito por rebolos ou lixadeiras utilizadas por meio de um processo manual, ou seja, por operadores. Esse processo, além de apresentar um alto tempo de execução e a mão de obra ser algo que está cada vez mais custosa, não garante uma uniformidade das peças ou repetibilidade do processo. Além disso, devido aos grandes esforços físicos exercidos pelos operadores, a questão da ergonomia se torna mais um aspecto a ser considerado. Assim, a busca pela automação desse processo se torna cada vez mais necessária. Uma alternativa é a aplicação de robôs industriais para realizar tal tarefa devido a sua flexibilidade e custos menores quando comparados principalmente a máquinas ferramentas. Sua programação é feita de duas formas, *on-line*, onde o robô enquanto é programado não produz, além de possuir limitações para trajetórias complexas, e *off-line*, onde o robô não é necessário durante sua programação, mas o processo possui erros e considerações a serem analisadas. Ambos os processos necessitam de um programador, e se faz necessário tempos de *setup* da célula ou estação de trabalho. Esse trabalho tem como objetivo propor e comprovar uma série de métodos de cálculos e procedimentos para contribuir com a automação dessa programação e *setup* do processo usando um robô industrial. O método propõe o uso de um escâner para adquirir o formato atual da peça a ser trabalhada e um *spindle* para realizar a remoção das rebarbas. Assim, através de comparação dos dados escaneados com o modelo 3D da peça projetada, esses métodos propostos avaliam então as geometrias a serem usinadas ou retrabalhadas e então, geram as trajetórias da ferramenta do robô. Basicamente, o trabalho contribui na forma que essa sequência de métodos encontra a peça no espaço de trabalho do robô, identifica as geometrias, gera suas trajetórias e em seguida, usina a peça ou o lote de peças desejado.

Palavras chave: Usinagem, rebarbação, robôs, geração de trajetória, escaneamento, alinhamento, rebarbação automática, reconhecimento de rebarbas.

## Abstract

In general, the products go through various and different stages during its manufacture. One of these stage is the finish. In the finishing step, it can be cited the deburring process. Burrs are characterized as undesirable projections in part or product and these need to be removed. There are two major reasons for their removal, visual aspects or even mechanical aspects. Burrs may cause accumulation of stresses and forces which can lead to fatigue piece or rupture. Thus, there is a need for processes for its removal. Currently, this process is made by grinding wheels or sanders used through a manual process, or by operators. This process, as well as slow and costly because labor is expensive, does not guarantee a uniformity of parts or repeatability of the process. Also, due to the great physical efforts exerted by operators, the question of ergonomics becomes another aspect to be considered. Thus, the search for automation of this process becomes increasingly necessary. An alternative is to apply industrial robots to perform this task due to its flexibility and lower cost when compared particularly to machine tools. Its programming is done in two ways, on-line, where the robot while it is programmed does not produce, as well as having limitations for complex trajectories, and off-line, where the robot is not necessary for their programming, but the process has errors and considerations to be analyzed. Both processes require a programmer and is required setup time of the cell or workstation. This work aims to propose and approve a number of methods of calculation and procedures to contribute to the automation of this program and process setup using an industrial robot. The method proposes the use of a scanner to acquire the current shape of the workpiece and spindle to perform removal of burrs. Thus, by comparing the scanned data with the 3D model of the projected part, these proposed methods then evaluate the geometries to be machined or reworked and then generate the robot tool paths. Basically, the work contributes to the way that this sequence of methods is the part in the robot workspace, identifies the geometry, generates their careers and then plant the part or batch of parts desired.

Keywords: machining, grinding, robots, trajectory generation, scanning, alignment

## 1 Introdução

Os produtos, durante seu processo de fabricação passam por uma ou mais etapas, processos. Exemplo desses são fundição, injeção, usinagem, polimento, rebarbação, limpeza por jato de areia entre outros. Alguns desses processos são caracterizados como processos da fase de acabamento. É importante ressaltar que a fase de acabamento pode ser constituída de mais de um desses processos [1, 2].

Como exemplo, em uma peça fundida, um dos processos de acabamento pode ser a rebarbação. Esse tipo de processo de acabamento é feito muitas vezes de forma manual. Isso pode encarecer o produto e também, aumentar os tempos de produção [1, 2].

Rebarbação consiste em remover as rebarbas formadas principalmente nas extremidades a fim de garantir a geometria final da peça dentro de tolerâncias projetadas e aceitáveis [3, 4].

Rebarbas são definidas como projeções de materiais indesejáveis em regiões ou além da borda de uma peça. São frequentemente ocasionadas devido a deformação plástica presentes em processos de fabricação como fresamento, torneamento, furação ou como fissuras ou imperfeições nas regiões de fechamento de moldes de fundição e de injeção [2, 5].

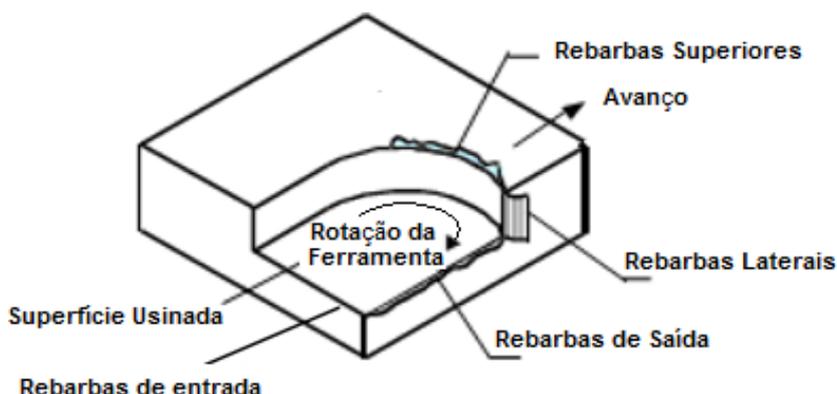


Figura 1.1: Formação de rebarbas em peças usinadas [2]

A remoção dessas rebarbas por si só, ou ainda, em materiais com alta resistência ou com características do tipo formatos que dificultam o acesso da ferramenta de rebarbação tornam muitos desses processos de acabamento tão complexos que podem ser caracterizados como inviáveis [1, 5].

Em alguns casos, as rebarbas são difíceis de serem detectadas, mas principalmente difíceis de serem mensuradas devido às suas formas e dimensões variáveis. A remoção de rebarbas chega a ser um gargalo na produção [6, 7].

São basicamente duas as principais aplicações do processo de rebarbação, qualidade superficial ou exigência mecânica [8, 9].

Como exemplo de aplicação para atender qualidade superficial, exigências estéticas, pode ser citado uma roda de alumínio. Essas, após a fundição, precisam ser necessariamente rebarbadas e polidas antes do processo de cromagem. Caso contrário, toda e qualquer parte que não foi adequadamente suavizada ficará visível como um defeito na peça após essa operação de cromagem [8, 9].

Como exemplo de aplicação para atender exigências de resistência mecânica, pode ser citado componentes de motores aeronáuticos, os quais são submetidos a altos níveis de solicitações dinâmicas como vibrações durante sua operação. A integridade mecânica desses componentes usinados ou fundidos pode ser comprometida pela presença de rebarbas ou arestas cortantes, pois esses causam acúmulo de tensão nos componentes. Dessa forma, a remoção de rebarbas é necessária para melhorar a resistência à fadiga térmica e mecânica desses componentes submetidos a um alto nível de estresse mecânico [8, 9].

A Figura 1.2 ilustra uma roda obtida pelo processo de fundição e suas rebarbas.

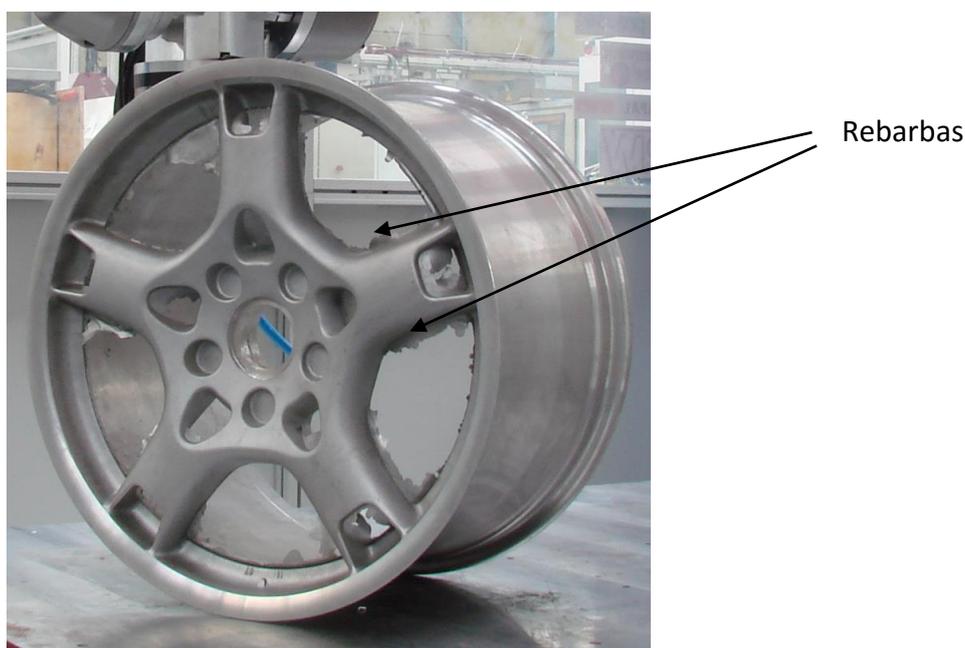


Figura 1.2: Roda de alumínio obtida pelo processo de fundição

Como exemplo, peças obtidas pelo processo de fundição, contém rebarbas, canais de enchimento, drenos entre outros. Alguns deles são necessidades do processo como os canais de enchimento e dreno, e outros são provenientes de uma não uniformidade do fechamento do molde como as rebarbas. Esses não pertencem a peça e portanto, precisam ser retirados [2, 10, 11].

Operações de limpeza e pré-usinagem são as principais atividades e representam um alto custo para as empresas. As operações de limpeza são responsáveis por cerca de 20% a 40% dos custos da manufatura por fundição. Além dos custos de mão de obra, pode-se destacar que as operações de rebarbação são realizadas na maioria das vezes em um ambiente com elevado nível de ruído, empoeirado e insalubre, tornam a rebarbação manual um processo caro e por isso muitas vezes inviável, ou seja, se faz para atender necessidades estéticas ou mecânicas do produto. Como uma alternativa viável de automação do processo, um robô industrial pode ser utilizado para remover as rebarbas de forma rápida e mais completa, e com melhor exatidão dimensional do que os processos manuais. Operações de limpeza, usinagem, rebarbação e corte são aplicações promissoras para robôs industriais com a tendência da automação da indústria de fundição [2, 10, 11].

Tradicionalmente, rebarbação tem sido uma operação manual que é muito trabalhosa, ineficiente, com o tempo de processo longo e de alto custo, sujeito a erros, e perigoso devido à poeira abrasiva [3].

Além desses fatores abordados, segundo uma visita a uma empresa de fundição na região de Santa Bárbara D'Oeste, em São Paulo, além do ambiente com elevado nível de ruído, empoeirado e insalubre, a rebarbação é também um processo que exige muito esforço físico dos trabalhadores, uma vez que essa produz peças de proporções elevadas e para seu acabamento, utiliza também de lixadeiras manuais. Segundo essa empresa, uma inviabilidade atual de se tentar automatizar principalmente com o uso de robôs, é a diversidade de peças que produzem e, portanto, o sistema teria que ser simples e rápido de configurar. O que não ocorre através de métodos *on-line* e ou até *off-line* de programação.

A indústria da manufatura está sempre buscando aprimoramento dos processos de produção. Requer cada vez mais processos produção modernos, que combinem otimização e flexibilidade. O tempo de produção total deve ser minimizado para assegurar o máximo de produção possível [12].

Muitos trabalhos estão sendo realizados na área de programação de robôs abordando programação *on-line*, *off-line*, usando câmeras de visão, células de carga entre outros buscando aprimorar e melhorar o uso de robôs na indústria.

Um dos objetivos de trabalhos e pesquisas nessa área, são métodos para a programação automática do robô, isto é, sem o uso de operador e ou programador. O objetivo da programação automática é minimizar as limitações decorrentes dos critérios usados pela programação através de um programador [13].

A aplicação de robôs vem se expandindo cada vez mais [14]. A procura pela automação desses processos, principalmente usando robôs de seis eixos vem se intensificando ano a ano. Principalmente com o uso de ferramentas para usinagens leves entre outros. Robôs para acabamento são utilizados tanto para operações mais leves, quando para operações mais pesadas como remoção de rebarbas [1].

Para a programação de robôs industriais, destacam-se dois principais métodos. *On-line* e *Off-line*. O *On-line* consiste em programar o robô levando a sua ferramenta ou órgão terminal para as sucessivas posições desejadas usando o *Teach Pad*, controle do próprio robô e armazenando o programa em sua memória. Nesse método, o robô não pode produzir durante a sua programação, pois o mesmo é usado para se auto-programar. O método *off-line* consiste em inserir o modelo matemático do robô e a planta que o mesmo vai atuar em um sistema de simulação e nesse fazer sua programação, restando apenas alguns ajustes para fazer no próprio robô [15, 16].

Dessa forma, a programação *on-line* ou geração manual da trajetória do robô é feita normalmente por operadores que movem o robô para sucessivas posições e orientações baseados no método de tentativa e erro. Além disso, esse método de programação *on-line* necessita mover o robô fisicamente, e, portanto, durante sua programação o mesmo não está produzindo. Pode-se ainda citar que nessa programação de rebarbação, assume-se que após a passagem da ferramenta do robô pelas áreas indicadas, a peça está perfeita, não havendo um retorno de informação sobre a superfície e nenhuma comparação da peça rebarbada com o projeto inicial da peça [9].

Portanto, métodos para a extração de dados numéricos como nuvens de pontos obtidos da peça a ser usinada ou rebarbada ou ainda, obtidos após a sua usinagem ou rebarbação e então o processamento desses dados poderão contribuir com essas operações através de diagnósticos automáticos principalmente quanto às geometrias e tolerâncias dimensionais da peça [17].

Um método de diagnóstico é a inspeção através do escaneamento da peça a ser trabalhada ou já trabalhada a fim de gerar nuvens de pontos. Em geral, o processo de comparação do modelo escaneado com o modelo CAD consiste basicamente em calcular a distância entre esses modelos. Antes de realizar esse processo, as nuvens de pontos tanto do modelo escaneado quanto ao modelo CAD da peça necessitam de estar alinhadas por processos de comparação. O mais comumente utilizado é o ICP – *Iterative closest point*. [18].

Dessa forma, o objetivo deste trabalho é o desenvolvimento de um método e sua implementação computacional para identificar as regiões com rebarbas e seu volume em uma peça fundida, injetada ou usinada e gerar automaticamente um programa NC para sua remoção com o auxílio de um cabeçote fresador montado em um robô. Esse objetivo é apresentado com mais detalhes no Capítulo 3.

#### Estrutura do Trabalho

Esse trabalho foi estruturado em 7 capítulos, sendo estes:

Capítulo 1: **Introdução** – Apresenta um panorama geral sobre rebarbação, sua remoção e como o trabalho pretende contribuir com o processo.

Capítulo 2: **Revisão Bibliográfica** – Apresenta uma revisão bibliográfica sobre o uso e a importância de robôs, fundamentos matemáticos necessários para a realização desse trabalho e os métodos aplicados.

Capítulo 3: **Objetivos e Metodologias** – Apresenta os objetivos deste trabalho e a metodologia utilizada para alcançar os objetivos propostos.

Capítulo 4: **Apresentação Geral do Conceito** – Nesse capítulo está descrito todo o conceito.

Capítulo 5: **Implementação e Validação** – Esse capítulo apresenta a implementação prática, e demonstra a sua validação e descrição dos experimentos, assim como os resultados obtidos.

Capítulo 6: **Conclusões** – Apresenta as conclusões obtidas por intermédio das pesquisas e da realização do método de reconhecimento de rebarbas.

Capítulo 7: Referências Bibliográficas

## 2 Revisão Bibliográfica

Nesse capítulo é apresentado uma revisão da literatura sobre os principais conhecimentos para a realização desse trabalho, assim como a sua inserção e sua diferença dos trabalhos já existentes na área.

### 2.1 Robôs industriais

O trabalho se baseia no uso de robôs industriais para a manipulação de um *scanner 3D* e de um *spindle* para a usinagem. Dessa forma, nesse item descreve o conceito de robôs industriais assim como alguns de seus principais aspectos.

#### 2.1.1 Histórico

O ser humano sempre buscou desenvolver e construir ferramentas e utensílios com o objetivo de auxiliar a realização das atividades do cotidiano. O conceito de evolução humana, principalmente para a civilização ocidental está diretamente relacionada com o grau de desenvolvimento tecnológico associado ao aperfeiçoamento dessas ferramentas e utensílios [19].

Em 1769, James Watt desenvolveu a máquina a vapor. Com esse novo conceito, houve progressos significativos nos processos produtivos. Esse conceito permitiu que máquinas realizassem tarefas pesadas e deu um grande impulso no desenvolvimento de tecnologias para auxiliar o homem [19].

Em 1923 Karel Capek utiliza o conceito do termo Robô. A palavra tem origem tcheca, *robotnik*, que significa servo. Abrange qualquer máquina que tenha a capacidade de auxiliar ou substituir o homem na realização de suas atividades. O desejo de se construir robôs não é só do homem moderno. Há relatos por exemplo de animações mecânicas feitas por Leonardo da Vinci [20].

Foi no início do século XX que robôs tomaram força. Os principais objetivos eram aumentar a produtividade industrial e melhorar a qualidade dos produtos [20].

Como fatores que beneficiaram o desenvolvimento dos robôs industriais, pode ser citado que na década de 60, quando foram lançados os primeiros robôs industriais, seus preços eram muito altos e somente pouquíssimas empresas de países desenvolvidos como Estados Unidos e Japão conseguiam acesso a essas máquinas. A partir de 1976, os preços começaram a baixar de forma acelerada impulsionando a tecnologia e facilitando o seu acesso. O principal argumento para essa redução é a microeletrônica. Com o avanço dos computadores e a redução

dos tamanhos dos circuitos para pequenos circuitos integrados alavancaram essa acessibilidade [21, 22, 23].

Podem ser citados alguns fatos históricos no desenvolvimento da robótica, esses são baseados nas referências [16, 19, 20, 21, 22, 23]:

- Século IV a.C (Grécia) – Aristóteles relata os primeiros princípios da robótica. Ou seja, o que é robótica e como se caracteriza.
- Século XVIII – Inicia-se a revolução industrial. Surgem novos conceitos industriais que permitiram que as máquinas controlassem ações sequenciadas.
- Século XIX – O motor elétrico é introduzido na indústria. Acontece um desenvolvimento significativo das máquinas com a introdução dessa nova tecnologia.
- 1914 a 1918 – A primeira guerra acarreta mudanças. As máquinas mostram que não trazem só benefícios a humanidade. Por exemplo, questões como desemprego e necessidade de treinamento de pessoal.
- 1921 – Karel Capek usa pela primeira vez a palavra *Robot* na peça teatral *Rossum's Universal Robots*.
- 1928 – Em Londres, um robô mecânico abre uma exposição de modelos técnicos.
- 1940 – Isaac Asimov estabelece quatro leis da robótica:
  - 1 Um robô não pode ferir um ser humano ou permanecer passivo ao perigo.
  - 2 Um robô deve obedecer às ordens dadas pelos humanos contanto que não contrariem a primeira lei.
  - 3 O robô deve proteger sua existência na medida a não contrariar a primeira e segunda lei.
  - 4 O robô não pode causar mal a humanidade e nem permitir que ela própria a faça. (Essa quarta lei foi escrita somente em 1984).
- 1954 – George Devol desenvolveu o primeiro robô programável.
- 1955 – Denavit e Hartenberg desenvolveram matrizes de transformação homogênea para cálculos de cinemática de robôs.

- 1959 – Devol e Joseph F. Engelberg desenvolveram o primeiro robô industrial pela Unimation Inc. capaz de executar automaticamente uma variedade de tarefas.
- 1960 – Torna-se significativo o fato da flexibilidade das máquinas. Começa o avanço para o tema robótica móvel.
- 1962 – H.A.Ernst inicia o desenvolvimento de um computador capaz de controlar uma mão mecânica.
- 1963 – American Machine & Foundry Company introduz no mercado uma versão comercial de robô.
- 1968 – McCarthy e outros, desenvolvem um computador com mãos, olhos, pernas e ouvidos demonstrando a capacidade de identificação, reconhecimento e manipulação de blocos espalhados sobre uma mesa.
- 1969 – Homem pisa no solo lunar. Foram usados manipuladores para recolher amostras e executar pequenas tarefas.
- 1970 – Inicia a pesquisa do uso de sensores nas operações.
- 1971 – Kahn e Roth analisam a dinâmica no controle de braços.
- 1973 – Em Stanford, Balles e Paul utilizam sensor visual e de peso em um braço robótico.
- 1974 – Cincinnati Milacron introduz o primeiro robô industrial controlado por computador denominado T3.
- 1975 – Will e Grossman desenvolvem na IBM um manipulador para a montagem mecânica de uma máquina de escrever.
- 1978 – O primeiro Robô Puma foi enviado a GM.
- 1980 – A General Motors, em Detroit, introduz um robô industrial com inteligência eletrônica.
- 1982 – A GM e Fanuc Japão assinaram um acordo para a construção de robôs GM Fanuc.
- 1983 – A Unimation foi vendida a Westinghouse, e posteriormente a vendeu a Staubli da Suíça em 1988.
- 1986 – A Honda introduziu o seu primeiro robô humanoide chamado de H0.
- 2005 – Entre Janeiro e Março, mais de 5300 robôs foram encomendados pelas empresas norte-americanas.

### 2.1.2 Características Quanto ao Uso da Robótica Industrial

De acordo com os autores [16, 19, 20, 21, 22, 23] podem ser citadas algumas características sobre robôs industriais.

- A automação possibilita incrementos na produtividade, através da realização de tarefas mais rapidamente do que usando processos manuais.
- Melhoram a qualidade dos produtos, uma vez que a repetibilidade é maior.
- Evita-se perda e refugos, pois robôs realizarão as tarefas sempre conforme programado e dentro de suas limitações operacionais.
- Elimina tempo morto – 24 hrs por dia trabalhando, pois, robôs trabalham continuamente, sem cansaço, fadiga ou tédio.
- Permite operação em ambiente difíceis e perigosos ou tarefas desagradáveis ou repetitivas.
- Robôs não precisam de condições ambientais como ar condicionado, luz e silêncio, fatores que influenciam no rendimento de tarefas manuais.
- Robôs principalmente não tem problemas emocionais que podem influenciar na produção ou segurança do trabalho.
- Quanto aos custos, robôs estão cada vez mais baratos e acessíveis. Porém, seu custo não envolve só o equipamento, envolve também:
  - Compra, que é o custo de aquisição do robô em si.
  - Instalação, que são os custos de preparação da célula robótica, como fundação, alimentação de energia elétrica entre outros.
  - Manutenção, que são os custos operacionais do robô, como manutenções periódicas e quebras de componentes.
  - Periféricos. Robôs por si só não realizam tarefas de forma mais eficiente. Alguns periféricos têm que ser instalados como esteiras alimentadoras, dispositivos de fixação em máquinas por exemplo que aceitem o robô as alimentando, ou seja, dispositivos que constituam de uma linha de produção ou célula robótica.
- Algumas questões a serem analisadas quando se desejar automatizar uma linha ou similar quanto a questão de retorno financeiro:
  - Número de empregados substituídos por robôs,
  - Número de turnos realizados por dia

- A produtividade comparada ao seu custo
- Custo de projeto e manutenção
- Custo de equipamentos periféricos

Esses itens são importantes, pois a automação só se justifica quando se números grandes de lotes de peças. Quando se deseja produzir poucas peças ou até peças unitárias, assim como se o produto não ficará tempo suficiente no mercado, a automação não justifica seus custos.

- Quanto a impacto social, robôs a curto prazos trazem problema como desemprego e treinamento de pessoal. Porém quando analisados ambientes perigosos, ou repetitivos, esses se tornam positivos na substituição dos humanos.

### 2.1.3 Conceitos Básicos de um Robô Industrial

A *Robotic Industries Association* (RIA), define robô industrial como um manipulador multifuncional reprogramável. Esse pode movimentar peças e componentes, ferramentas ou outros através de movimentos programados para diversas tarefas [16].

A Figura 2.1 ilustra um robô industrial de 6 eixos, portanto, seis graus de liberdade. Os nomes dos eixos ou movimentos são baseados no braço do ser humano [16].

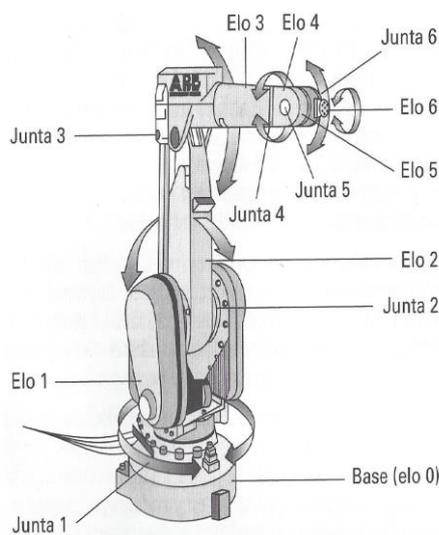


Figura 2.1: Robô Industrial de 6 graus de liberdade [16]

O número de juntas determina os graus de liberdade do robô. A maioria deles possui de 3 a 6 eixos os quais podem ser divididos em duas classes: Eixo do corpo

e Eixo da extremidade. Os eixos do corpo permitem movimentar a ferramenta terminal para uma determinada posição no espaço, são denominados cintura, ombro e cotovelo (*waist*, *shoulder* e *elbow*). Os eixos da extremidade, denominados rolamento, mergulho e guinada (*roll*, *pitch* e *yaw*) permitem orientar a ferramenta terminal [16, 20].

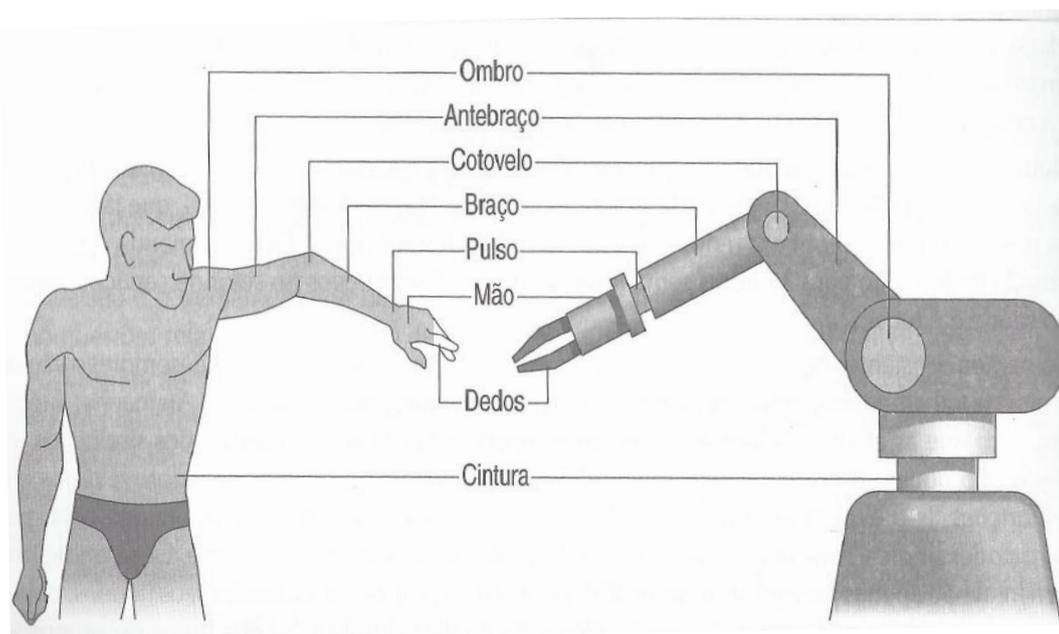


Figura 2.2: Associação ao braço humano [20]

Como componentes de um robô, pode ser citado [16, 20]:

- Manipulador: Corpo principal do robô, que consiste nos elementos estruturais do robô.
- Efetuador ou atuador final: São basicamente o último elo do robô. Podem ser classificados como ferramentas especiais ou garras mecânicas. Tem a função de realizar a ação.
- Atuadores: São os “músculos” do robô. São os motores ou atuadores pneumáticos ou hidráulicos que fazem a movimentação mecânica dos elos.
- Sensores: São usados para coletar informações internas do robô ou ainda do meio externo.
- Controlador: É responsável por controlar os movimentos,
- Processador: Responsável por calcular os movimentos.

- Software: Existem basicamente 3 grupos. O primeiro é o operacional, o segundo é o que calcula os movimentos baseado na cinemática e o terceiro é o que controla as rotinas de programação.

#### 2.1.4 Classificação de robôs por quanto a estrutura mecânica

O objetivo desse item é através de um estudo sobre os principais tipos de robôs, encontrar o qual apresenta melhores benefícios para o uso no projeto.

Basicamente são divididos em cinco tipos, que serão caracterizados a seguir [19]:

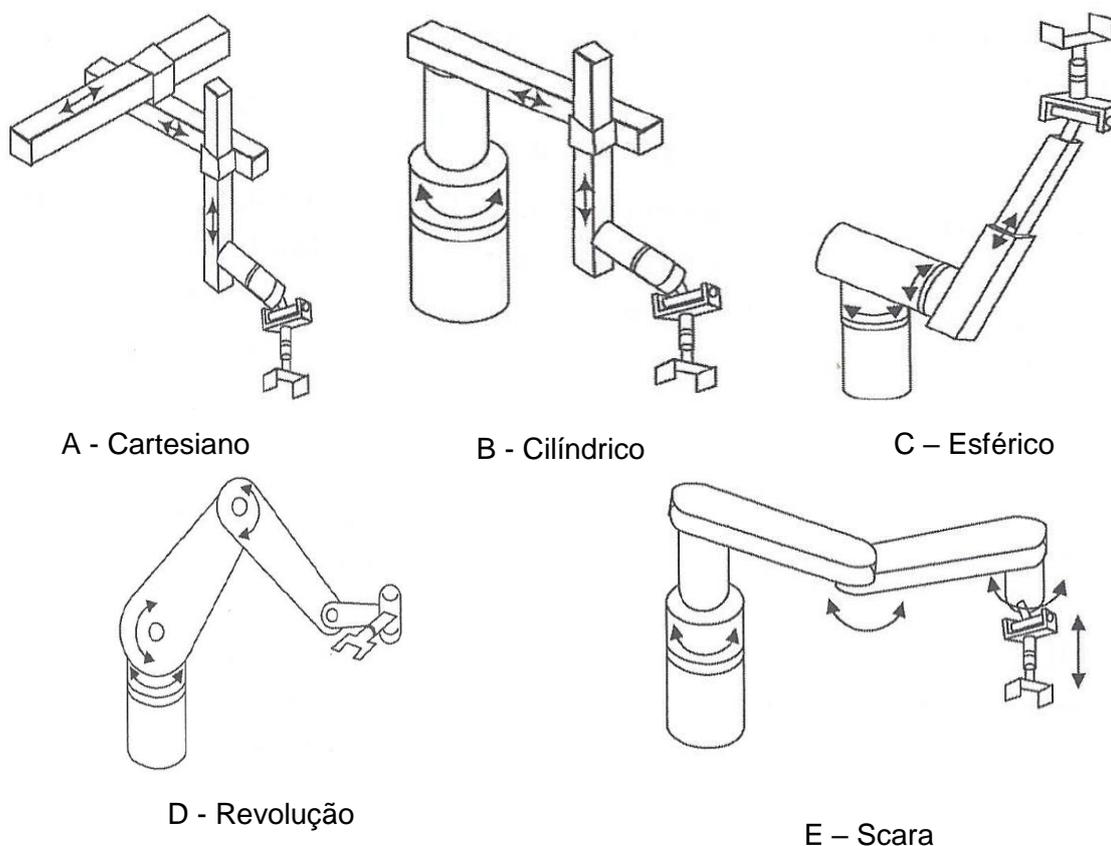


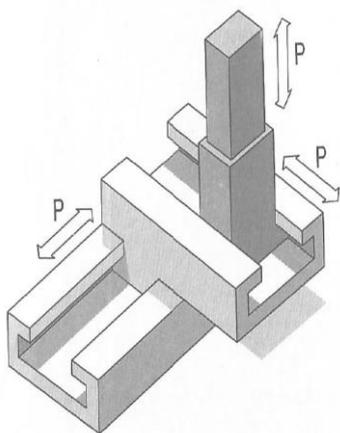
Figura 2.3: Classificação de Robôs quanto a estrutura mecânica [19]

A seguir, segue um comparativo dentre as características dos cinco tipos de classificação. Esse comparativo é entre eles.

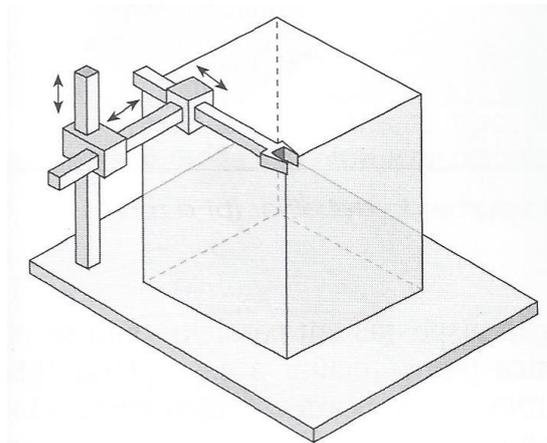
- Robô de coordenadas cartesianas ou pórtico
  - Possui três juntas prismáticas (PPP).
  - Maior grau de rigidez
  - Controle mais simples

- Seu volume de trabalho é um paralelepípedo e possui o menor volume de trabalho

Ou seja, é considerado o tipo de robô mais rígido, seu controle é mais simples, porém, apresenta o menor volume de trabalho dentre os cinco tipos.



Cartesiano A



Cartesiano B

Figura 2.4: Representação do robô tipo cartesiano onde: A – Representa a esquemática do robô cartesiano [20]. B- Representa o volume de trabalho do robô cartesiano [16].

- Robô de coordenadas cilíndricas
  - Possui uma junta rotacional e duas juntas prismáticas (RPP).
  - Sua rigidez mecânica é menor do que a do cartesiano.
  - Controle um pouco mais complexo devido a presença de uma junta rotacional.
  - Seu volume de trabalho é um cilindro e maior do que a do cartesiano.

Ou seja, é considerado o tipo de robô menos rígido quando comparado ao cartesiano, menos simples quanto ao seu controle, porém apresenta um volume de trabalho maior do que o cartesiano e menor que os demais tipos de robôs.

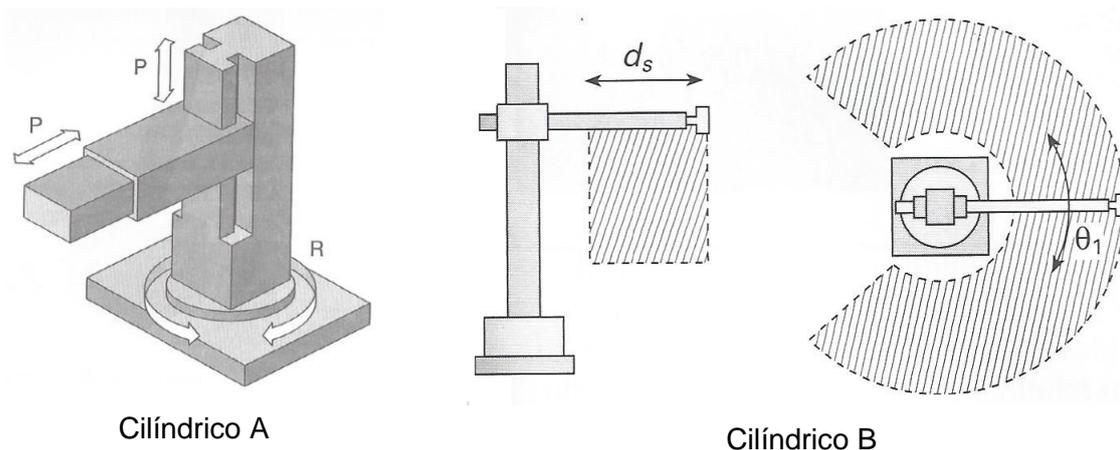


Figura 2.5: Representação do robô tipo cilíndrico onde: A – Representa a esquemática do robô cilíndrico [20]. B- Representa o volume de trabalho do robô cilíndrico [16].

- Robô de coordenadas polares ou esféricas
  - Possui duas juntas rotacionais e uma junta prismática (RRP).
  - Sua rigidez mecânica é ainda menor que a do cilíndrico.
  - Controle ainda mais complexo devido a presença de duas juntas rotacional.
  - Seu volume de trabalho é aproximadamente uma esfera e é maior do que a do cilíndrico.

Ou seja, é considerado o tipo de robô ainda menos rígido do que os já apresentados, ainda mais complexo quanto ao controle, porém apresenta o volume de trabalho maior do que os já apresentados.

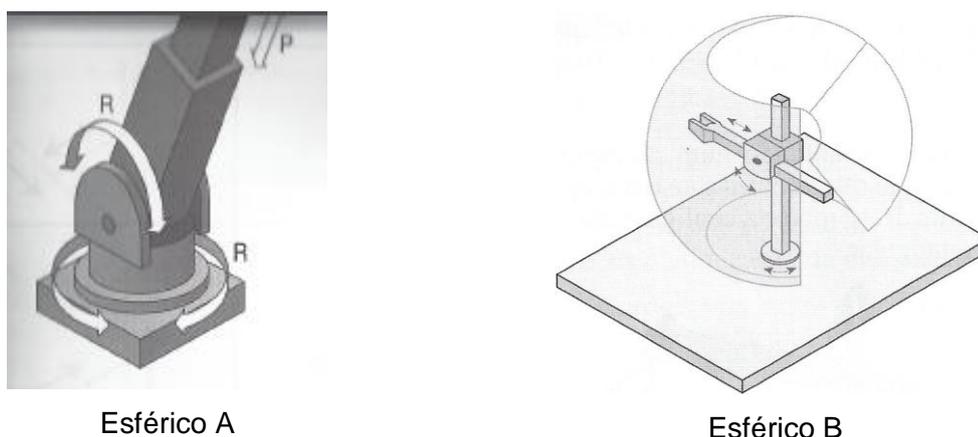
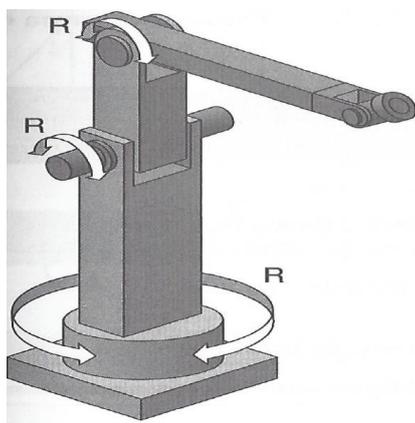


Figura 2.6: Representação do robô tipo esférico onde: A – Representa a esquemática do robô esférico [20]. B- Representa o volume de trabalho do robô esférico [16].

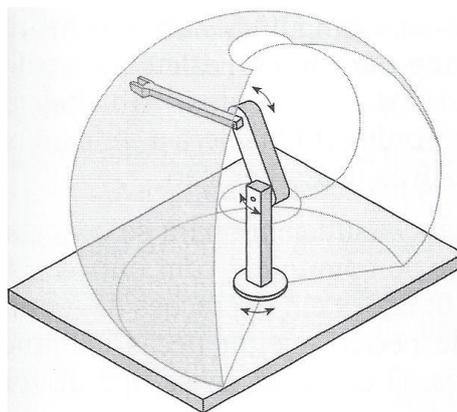
- Robô de coordenadas de revolução

- Possui três juntas rotacionais (RRR).
- Sua rigidez mecânica é ainda menor que a do cilíndrico.
- Controle é o mais complexo devido à presença apenas de juntas de revolução.
- Seu volume de trabalho é o maior de todos e sua forma é complexa.

Ou seja, é considerado o tipo de robô menos rígido, o mais complexo para controlar, porém, possui o maior volume de trabalho.



Revolução A



Revolução B

Figura 2.7: Representação do robô tipo revolução onde: A – Representa a esquemática do robô revolução [20]. B- Representa o volume de trabalho do robô revolução [16].

- Robô do tipo Scara
  - Possui duas juntas rotacionais e uma junta prismática (RRP).
  - Sua rigidez mecânica pode ser comparada ao cilíndrico.
  - Controle é complexo, porém menos do que o de revolução.
  - Seu volume de trabalho é um cilindro.
  - Sua principal aplicação é em operações *Pick and Place*. Devido ao fato do eixo Z ser uma junta prismática, seu controle se torna mais simples e sua velocidade é elevada para esse tipo de operação.

Ou seja, suas características podem ser comparadas ao cilíndrico, mas devido ao seu aspecto construtivo, se faz muito eficiente para operações de manejo de produtos e peças.

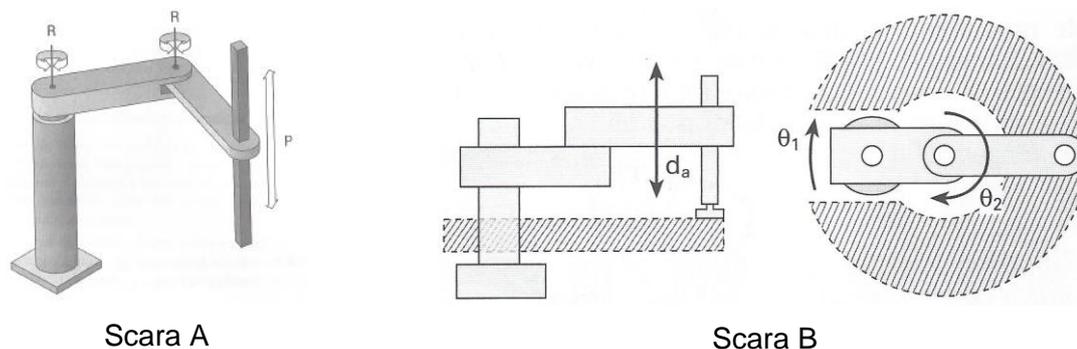


Figura 2.8: Representação do robô tipo scara onde: A – Representa a esquemática do robô scara [20]. B- Representa o volume de trabalho do robô scara [16].

Definição de ângulos de orientação RPY:

Conforme apresentado na Figura 2.9, a orientação da ferramenta ou órgão terminal é dada por três orientações:

- *Roll*: Movimento de rotação em torno do punho.
- *Pitch*: Movimento de rotação do punho para cima e para baixo.
- *Yaw*: Movimento de rotação do punho para a direita e para a esquerda.

Os movimentos de rotação *Roll*, *Pitch* e *Yaw*, são traduzidos para o português como: rolamento, mergulho e guinada. Conforme pode-se observar na figura. Várias áreas além da robótica usam o mesmo método para referenciar orientação. Em aviação por exemplo, a sua orientação é similar à da robótica [20].

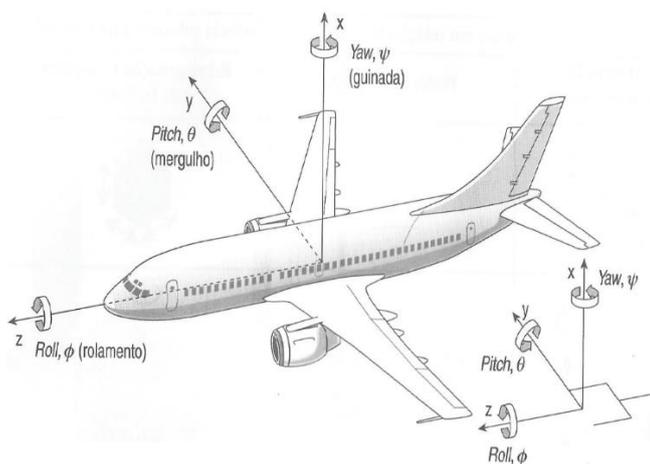


Figura 2.9: Definição dos ângulos de orientação RPY [20]

### 2.1.5 Sistemas de referência de um robô:

Os robôs podem ser referenciados em relação a diferentes sistemas de coordenadas. Sendo que em cada uma delas, a forma de movimentar o robô como

um todo, seus eixos ou a ferramenta ou órgão terminal é deferente. Normalmente, são três os sistemas de referência:

- Sistema de referência global: É o sistema universal de coordenadas definidos pelos eixos X, Y e Z na base do robô. Nesse sistema, não importa a posição ou orientação do braço, um movimento ao longo do eixo X no sentido positivo é sempre positivo em relação ao eixo X do referencial de base. É usado principalmente em relação a outros objetos ou máquinas que o robô se comunica e para definir trajetórias de movimento.

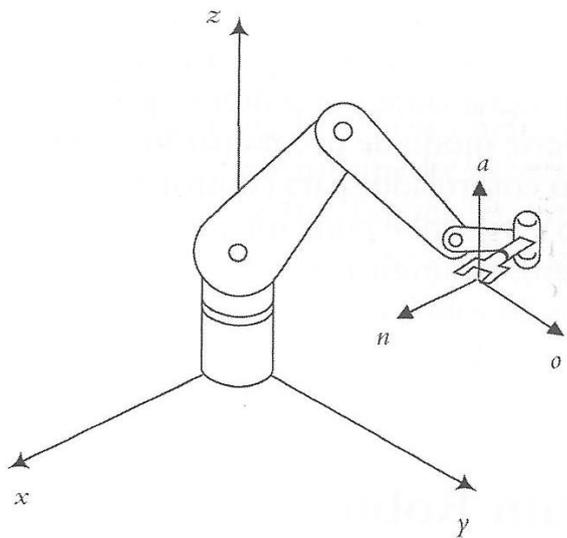


Figura 2.10: Referência Global [19]

- Sistema de referência das articulações: É usado para movimentar as articulações individuais do robô.

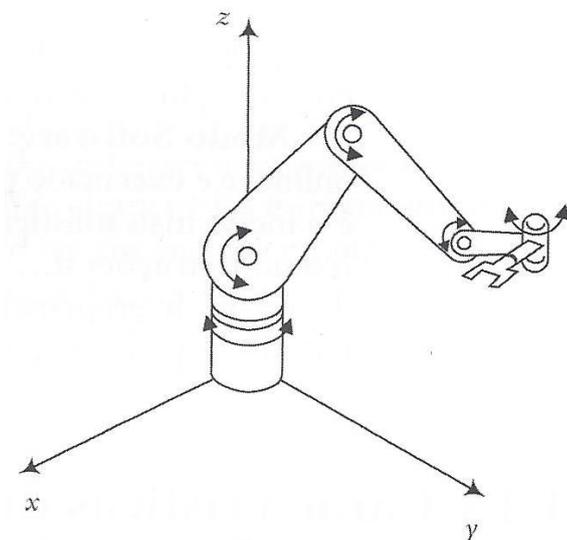


Figura 2.11: Referência Articulação [19]

- Sistema de referência ferramenta: São referentes ao sistema de referência da mão, ou órgão terminal do robô. Ao contrário do sistema global. Supondo que a mão está na posição indicada pela Figura 2.10, movendo a mão no sentido do eixo N positivo, a mão se moverá no sentido X positivo em relação ao sistema global. Supondo agora a Figura 2.12, movendo a mão no sentido do eixo N positivo, ele estará se movendo no sentido do eixo X negativo. Ou seja. Nesse sistema de coordenadas, o sistema de referência se modifica continuamente dependendo da posição do robô. É usado muito para a aproximação da mão ou ferramenta ou órgão terminal de peças e objetos [19].

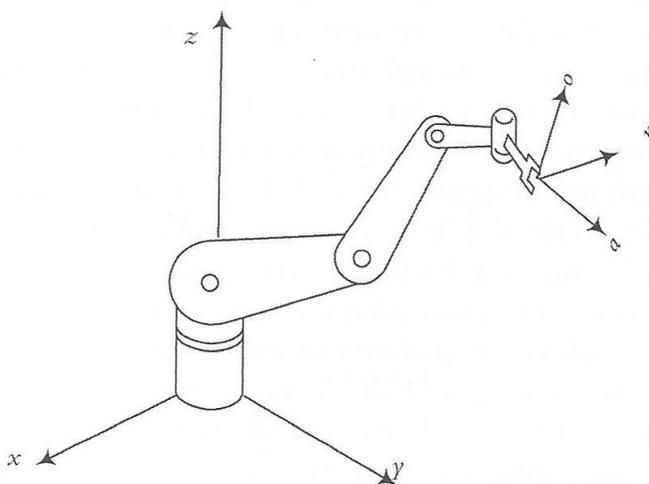


Figura 2.12: Referência Ferramenta [19]

É importante ressaltar que a orientação da ferramenta ou órgão terminal se dá por três vetores unitário denominados de N, S e A. São similares a eixo X, Y e Z, apenas para diferenciar que são da ferramenta ou órgão terminal levam esses nomes.

### 2.1.6 Programação de um Robô Industrial

Robôs são mecanismos automatizados projetados para movimentar peças ou ferramentas sobre trajetórias previamente estabelecidas.

A programação pode ser feita de duas maneiras: *On-line* e *Off-line*. Basicamente, na programação *on-line* se faz pelo método de ensino ou condução. No método *off-line*, são usados linguagens e programas de simulação. Na programação *on-line* se faz necessário o uso do robô e na programação *off-line*, a sua programação é desenvolvida parcialmente ou completamente sem a necessidade do uso do robô [16 20].

### 2.1.6.1 Programação *On-line*

É definido como programação on-line a utilização de métodos de programação “por ensino” para desenvolver e implementar um programa no controlador do robô. Basicamente é via *teach-pendant* ou com dispositivo mestre-escravo.

Esse tipo de programação é basicamente a inserção de pontos e suas modificações para definir as trajetórias e também, funcionalidades especiais como velocidade, repetição ou ainda, ativação de sinais de entrada ou saída através de I/Os.

Basicamente os algoritmos de controle são três:

- Ponto a ponto: Nesse tipo, o robô se move de uma determinada posição para a próxima posição sem a necessidade de que posições intermediárias sejam inseridas.
- Movimento contínuo: O robô se movimenta através de pontos com pequenos incrementos entre si. As posições são gravadas a cada intervalo de tempo e suas coordenadas são armazenadas através das leituras dos sensores de posição de cada eixo.
- Controle de trajetória: envolve o controle coordenado de todas as juntas para percorrer a trajetória desejada entre dois pontos.

Como vantagem, o fato de estar usando diretamente o robô garante que as posições são exatamente onde se pretende colocar o robô e uma vez definido o programa, o robô está pronto para uso.

Como desvantagem, durante a programação o robô não estará trabalhando e produzindo, dessa forma, gerando gastos ao invés de lucros. Além desse, programas mais complexos, principalmente quando envolvem tarefas como usinagem ou manipulações complexas usando mais de um robô, se torna muito complicado mover robô a robô na linha. Isso porque em linhas com mais de um robô, eles na maioria das vezes estão um dentro do volume de trabalho do outro, e portanto, deve se mover os robôs com cautela de forma a não colidir os mesmos. Por isso, não se pode programar um robô por total e depois o próximo e assim por diante. Tem que ser simultâneo. E então, como esse processo leva tempo, ainda tem o problema de deixar toda a linha improdutiva enquanto se programa a mesma [16, 20].

A programação *on-line*, pode ocorrer basicamente de duas formas, são elas *teach-pendant* e por condução e são apresentadas a seguir:

### **Programação Via *Teach- pendant***

A programação *on-line* necessariamente faz o uso de um teclado portátil chamado de *teach-pendant*. Basicamente, nesse tipo de programação, através de um *joystick* no teclado, o programador “leva” o robô para sucessivas posições e as insere no programa. A cada escolha de um novo ponto na trajetória, o programador pode definir se a trajetória entre esses dois pontos será linear, circular ou ainda, a mais rápida possível. Além dessas informações, podem ser definidas ou redefinidas velocidade dessa trajetória e ainda, a espera de sinais externos ou o envio de sinais para o mundo exterior para a interação com outras máquinas e dispositivos [16, 20].

### **Programação por Condução**

Nesse tipo de programação, o programador move fisicamente a mão, ou ferramenta do robô pela trajetória desejada. Para trajetórias contínuas, a cada tempo o controlador lê as informações das posições dos sensores e as armazena. Para trajetórias ponto a ponto, o operador diz especificamente quando quer que o controlador faça a leitura das posições dos sensores [16, 20].

#### **2.1.6.2 Programação *Off-line***

Quando se tem linhas de montagem ou fabricação envolvendo diversos robôs, colisões são possíveis devido ao fato de que os robôs estão muito próximos um do outro. O fato de que os robôs estão muito próximos também implica em outro fator. A complexidade das trajetórias, pois muitas vezes um robô está na rota de colisão do outro. Item necessário quando vários robôs estão trabalhando em uma mesma peça e ainda, existe possibilidade de outros robôs estarem segurando a peça a ser trabalhada. A programação *off-line* se faz essencial quando temos essa situação [16, 20].

Como vantagens, tem-se:

- Redução do tempo ocioso: O robô pode continuar a trabalhar na linha de produção enquanto a nova tarefa é programada.
- Ambientes potencialmente perigosos: Redução do tempo de permanência do operador próximo ao robô.

- Sistema simplificado de programação: Pode se usar essa forma para programar uma variedade de robôs sem se preocupar com as peculiaridades de cada controlador.
- Integração com sistemas CAD/CAM: Possibilita outras funcionalidades como planejamento e controle.
- Depuração de programas: Sistemas com integração CAD/CAM permitem o uso de um modelo computadorizado da planta, robô mais célula de trabalho, que pode ser usado para detectar colisões melhorando ainda mais a confiabilidade na programação [16, 20].

Como desvantagens, tem-se:

- Dificuldade na determinação precisa dos objetos. Robô, máquinas, peças com relação ao sistema de coordenada global do robô.
- Alterações no ambiente como temperatura podem alterar o desempenho do robô.
- A qualidade dos dados do modelo real.

## **2.1.7 Aplicabilidades de robôs industriais**

Robôs são amplamente utilizados em diversas áreas. A seguir, seguem tópicos sobre o uso de robôs nas áreas de montagem e manipulação, soldagem, pintura e usinagem.

### **2.1.7.1 Aplicabilidades na Montagem e Manipulação**

Operações de transporte e manuseio de materiais (*“materials handling”*) são operações responsáveis por transportar materiais, peças ou partes de uma posição para outra. Exemplo, de uma esteira para uma máquina, ou dispositivo de fixação entre outros. É uma aplicação consolidada e ideal para robôs devido à pouca complexidade da tarefa e pela questão de ser repetitiva [15].

Em operações que requerem repetibilidade e tolerâncias dimensionais, robôs industriais mostram vantagens em relação ao trabalho manual [24].

Os processos de manipulação e montagem representam cerca de 50 a 70% dos custos de mão de obra de um produto. Dessa forma, é o processo onde envolve maior número de pessoas no processo de manufatura. Dessa forma, robôs se mostram extremamente promissores para a aplicação nesse processo [16].

Basicamente, processos de montagem podem ser agrupados em três categorias: Manual, mecanizada e robotizada [16].

Basicamente, os usos de montagem robotizada indicam que:

- O setor automotivo é o que concentra o maior número de aplicações, seguido da indústria eletrônica.
- Há cerca de dez peças singulares por produto.
- 80% das peças a serem montadas pesam menos de 4 kg
- 70% das dimensões internas das peças a serem montadas são semelhantes a um cubo com 100mm de lado.
- O tempo de ciclo de montagem é de 12 a 60s nas linhas de montagem e de 20s a 3 min nas células de montagem [16].

A Figura 2.13 apresenta um robô destinado à montagem e manipulação, da marca ABB, modelo IRB 6620.



Figura 2.13: Robô ABB IRB 6620 para manuseio 25

A Figura 2.14 apresenta um robô com funcionalidades de montagem e transporte, da empresa KUKA.



Figura 2.14: Robô KUKA KR 6-2 destinado à operação de manuseio [26]

### 2.1.7.2 Aplicabilidades na Soldagem

Um dos principais fatores com relação a soldagem é a questão da qualidade do cordão de solda com relação a homogeneidade da microestrutura e da característica física da qualidade da solda. Isso porque, quando o processo é manual, mesmo o soldado sendo experiente, esse não consegue ter repetibilidade em todos os cordões de solda. Robôs são uma excelente alternativa para automação desse processo e estão cada vez mais presentes na indústria para tal realização [27].

Em geral, o problema no emprego de robôs na soldagem está no trabalho adicional e principalmente inicial de programação e configuração para garantir que os mesmos atinjam a qualidade especificada. Conforme já citado, a maioria dos processos de programação são feitos por *teaching* ou programação *off-line*, que demanda tempo e pessoal qualificado [28].

O primeiro processo de soldagem robotizado foi por resistência elétrica, soldagem a ponto, na década de 70. Esse processo foi introduzido na indústria automobilística e mudou completamente o conceito de linha de produção [16].

Os processos mais usados para soldagem robotizada, aplicados principalmente por envolver operações altamente repetitivas são:

- GMAW, *Gas Metal Arc Welding*, também conhecido por MIG/MAG
- GTAW, *Gas tungsten arc welding*, também conhecido por TIG

- LBW, *Laser beam welding and cutting*, que é soldagem e corte a laser
- PAW, *Plasma arc welding and cutting*, que é soldagem e corte a plasma
- RSW, *resistance spot welding*, soldagem por resistência elétrica por pontos, que é a solda a ponto [16, 29].

Um dos principais problemas relacionados a soldagem é a qualidade do cordão de solda relacionada com a homogeneidade da microestrutura e suas características físicas e dimensionais. Esses fatores estão diretamente relacionados quando o processo é manual. Isso porque o soldador não é capaz de soldar com boa repetibilidade todos os cordões de solda. Dessa forma, robôs estão cada vez mais viáveis para esse tipo de tarefa, devido principalmente à sua boa repetibilidade [27].

Há alguns trabalhos na área de alta eficiência, controle digital e robotização na área da solda elétrica na direção da inserção da integração da inteligência artificial, visão computacional, processamento digital e robôs com alta tecnologia. Essa tendência visa principalmente possibilitar que o processo seja mais flexível e menos dependente do homem [28].



Figura 2.15: Robô ABB IRB 1600ID destinado a Soldagem [30]

A Figura 2.15 mostra o IRB 1600ID, da empresa ABB. Todos os cabos e mangueiras estão localizados dentro do braço, tornando o robô perfeitamente adequado para a soldagem.

### 2.1.7.3 Aplicabilidades na Pintura

Na área da área da pintura, robôs também são cada vez mais aplicados. Estão presente em diversas áreas, mas principalmente na indústria automotiva. Robôs se tornam uma alternativa viável e segura em produtos que exigem uniformidade da espessura da tinta pois esse fator influencia diretamente na qualidade do produto.

A qualidade de uma pintura está diretamente ligada a trajetória da ferramenta de pintura. Quando essa ligada a um robô, esse tem boa repetibilidade da trajetória mantendo assim a qualidade desejada do produto. Atualmente, há dois métodos para o planejamento da trajetória para um robô de pintura: O manual e o automático [24, 31].

O método manual de planejamento da trajetória é realizado basicamente baseado nas experiências dos programadores e engenheiros de produção, instalação, equipamentos e seus conhecimentos de processos. Um engenheiro de produção tem que executar testes extensivos em uma célula de trabalho para otimizar um caminho de ferramenta para obter uma distribuição uniforme de pintura. Esse processo além de muito complexo, varia de engenheiro ou programador para engenheiro ou programador. Existem sistemas de planejamento da trajetória da ferramenta com auxílio de computador (CATP - *Computer aided tool path planning*), no qual a trajetória da ferramenta é automaticamente correlacionada com o modelo CAD da peça a ser produzida, o que reduz o trabalho humano e evita que os operadores sejam expostos a ambientes de trabalho prejudiciais. Porém, ainda é essencial desenvolver metodologias de planejamento da trajetória para substituir o processo manual pelo automático [31].

A Figura 2.16 apresenta o Robô IRB 52, da empresa ABB, destinado à pintura.



Figura 2.16: Robô IRB 52, destinado para pintura [32]

#### 2.1.7.4 Aplicabilidades na Usinagem

Robôs tem vantagens sobre máquinas ferramenta em relação à volume de trabalho, flexibilidade e custos de aquisição do equipamento. Dessa forma, nos últimos anos, robôs industriais estão sendo cada vez mais cogitados para a aplicação em operações de usinagem e processos de rebarbação. Vários estudos vêm sendo realizados sobre robôs articulados para aplicações que envolvem o conceito de usinagem [33, 34].

Um dos principais fatores de ainda serem poucos utilizados, é o fator da baixa rigidez mecânica. Robôs articulados apresentam baixa rigidez e apresentam deformações substanciais, reduzindo assim a precisão de sua trajetória [33]. Os principais fatores que influenciam no resultado da aplicação de robôs na usinagem são os relacionados à rigidez estática e dinâmica de seus componentes [35, 36]. Além dos fatores mencionado, existe ainda um outro fator que dificilmente é considerado no momento da aquisição do equipamento. É a questão da frequência de vibração da ferramenta, que é causada pela rotação da mesma e pelo seu toque na peça. Quando essa coincide com a frequência natural do braço robótico, esse pode variar muito sua tolerância de posicionamento [36].



Figura 2.17: Robô para Usinagem ABB IRB 6660 [38]

Atualmente, quando robôs são aplicados para operações de usinagem, são usadas geralmente ferramentas menores e mais leves. A remoção de material é realizada por uma ferramenta de corte rotativa enquanto o robô garante trajetórias em superfícies complexas ou em áreas de difícil acesso para máquinas convencionais [36].

A Figura 2.17 apresenta o IRB 6660, da ABB, para usinagem. É um robô da marca dedicado à usinagem. Este novo robô foi projetado para aplicações de alto desempenho, nos quais a estabilidade é um fator chave para o sucesso [37].

## **2.2 Sistemas para a automação do processo de programação e controle de um robô industrial**

Nesse tópico, são abordados trabalhos até então realizados para contribuir com a programação e ou controle de posição de sistemas robóticos voltados principalmente para o processo de usinagem e rebarbação.

### **2.2.1 Aplicações na substituição do trabalho manual complexo**

A principal função de um sistema robótico, assim como qualquer sistema mecânica ou elétrico é substituir o trabalho do homem. Realiza-lo mais rapidamente, com

maior repetibilidade, qualidade e até, com menores custos que o trabalho manual. Trabalhos considerados complexos também são alvos da automação, apesar de serem mais trabalhosos para serem implementados.

Como exemplo desse objetivo, o artigo de Uhlmann et al., 39, descreve que componentes com grande valor agregado ou longo tempo de vida útil, quando apresentam defeitos, não são eliminados, mas sim, reparados. Ou seja, é mais viável reparar esses componentes e coloca-los para trabalhar novamente do que produzir um novo. Porém, devido ao tamanho do lote de peças que se deseja trabalhar, reparar ser muitas vezes unitário, esse é feito de forma manual. O artigo descreve que aplicando um processo de usinagem ou retificação adaptativo automático a um braço robótico, esses se mostram uma máquina flexível para efetuar tal nível de aplicação [39].

### **2.2.2 Compensação de erros na usinagem com robôs**

Nos dias atuais, a grande maioria das aplicações usando robôs são em operações como por exemplo pintura, *pick and place*, que são movimentações de peças e componentes, soldagem entre outras. Essas operações requerem do robô basicamente boa repetibilidade e os esforços sobre o braço robótico são constantes e podem ser previsíveis no momento da programação da célula robótica. Porém, as aplicações em outras áreas que requerem essa mesma repetibilidade, mas o braço robótico sofre ainda grandes solicitações, e essas não são constantes vem crescendo. São exemplos dessas áreas as correlacionadas a usinagem, que envolvem a usinagem em si, furação, rebarbação entre outras. Maquinas ferramentas, apesar de sua precisão e rigidez mecânica serem maiores do que as de robôs, essas perdem quando comparado a preços e flexibilidade [34].

Diversos trabalhos estão sendo realizados na área para compensar os erros durante o processo de usinagem. Dentre esses erros, os mais citados são o fato do braço fletir de acordo com os esforços de corte, vibrações durante o processo de corte do material, erros de posicionamento do próprio robô ocasionados por erros de comando, mecanismos etc entre outros.

Schneider et al., [34], propõe um método onde através de medições de esforços, esse consegue saber as variações de posições sofridas pelo braço robótico e então, compensar através de um mecanismo próprio. Esse sistema de compensação pode compensar em todas as direções até meio milímetro. A Figura 2.18 ilustra o processo.



Figura 2.18: Robô e seu dispositivo para compensação de erros durante a usinagem [34]

O trabalho de Liao et al., [3], propõe uma *toolhead* com um *spindle* pneumático que pode se estender e retrair através de três atuadores pneumáticos; Através de um sensor de pressão, a ideia do cabeçote é manter a pressão da ferramenta sobre a peça constante tanto em operações de polimento quanto de rebarbação. A Figura 2.19 ilustra a proposta do artigo.

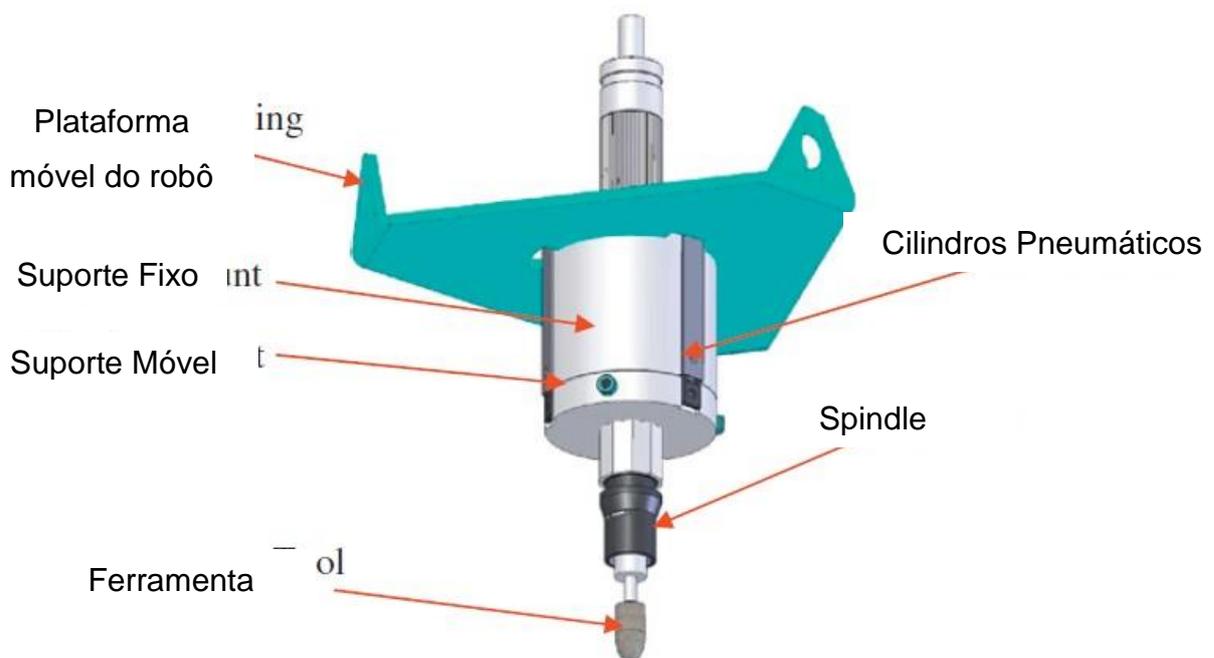


Figura 2.19: Cabeçote compensativo proposto por Liao et al. [3]

E a Figura 2.20 ilustra basicamente o fluxograma do processo.

Basicamente, o sensor de pressão na própria câmara do cilindro envia esses dados a um sistema no computador. Esse faz os cálculos de compensação e age na válvula proporcional de forma a estabilizar os esforços de corte. O trabalho relata que para os esforços de corte dos testes realizados o sistema se mostra funcional para auxiliar na correção de erros de posicionamento devido a esforços de corte.

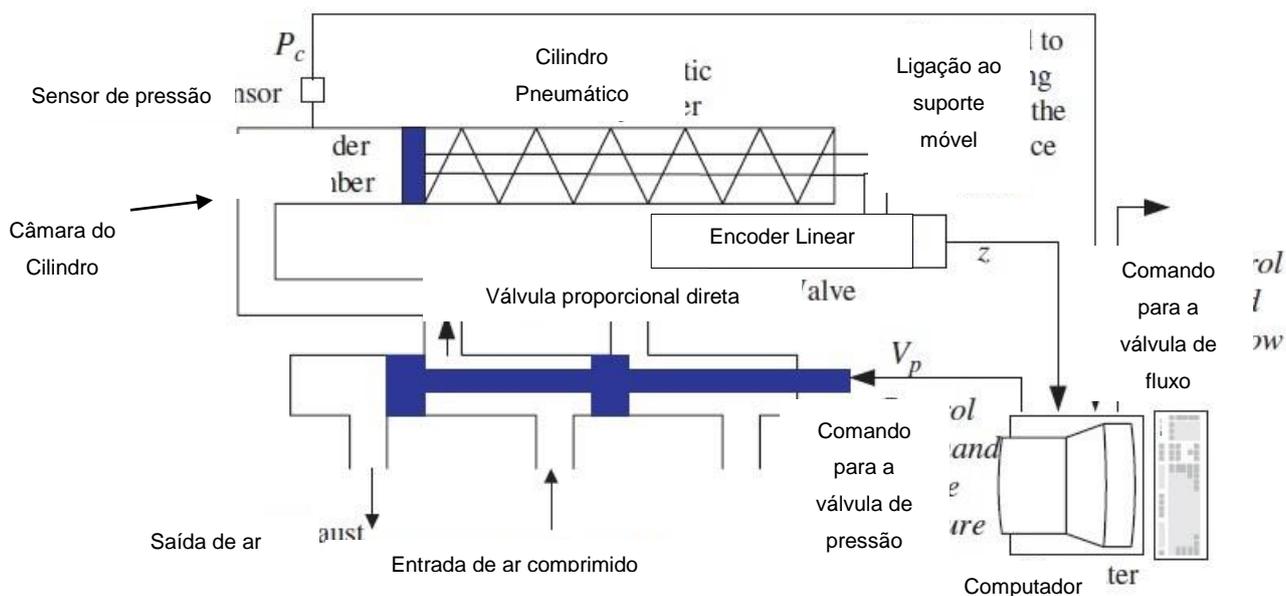


Figura 2.20: Fluxograma do processo do controle de torque pneumático [3]

Na mesma linha de pesquisa, tem-se o trabalho de Kim et al., [24]. Que também através de sensores de pressão, o sistema utiliza de atuadores pneumáticos com a intenção de manter o esforço da ferramenta sobre a peça. A Figura 2.21 ilustra o processo físico. A diferença principal entre esse trabalho e o anterior é que nesse a ferramenta é instalada no robô e o dispositivo compensador fica antes do *Spindle*, e, portanto, na mesa, fica apenas o dispositivo de fixação com a peça a ser trabalhada. Também se mostra eficiente a proposta segundo o artigo.

O trabalho de Ziliani et al., [40], propõe um controle de velocidade e força mecatrônico. Através de sensores eletrônicos e uma malha de controle, o trabalho propõe um controle híbrido do manipulador baseado na força e velocidade da ferramenta rebarbadora. A correção é feita através de envio de coordenadas de correção ao comando do robô para alteração da trajetória do robô. Uma vantagem desse processo é a não necessidade de um dispositivo externo ao robô para corrigir esses erros causados pelos esforços durante a usinagem. Uma

desvantagem é a necessidade de uma programação avançada do robô para alterar constantemente o programa a ser executado. A Figura 2.22 ilustra a malha proposta pelo artigo.

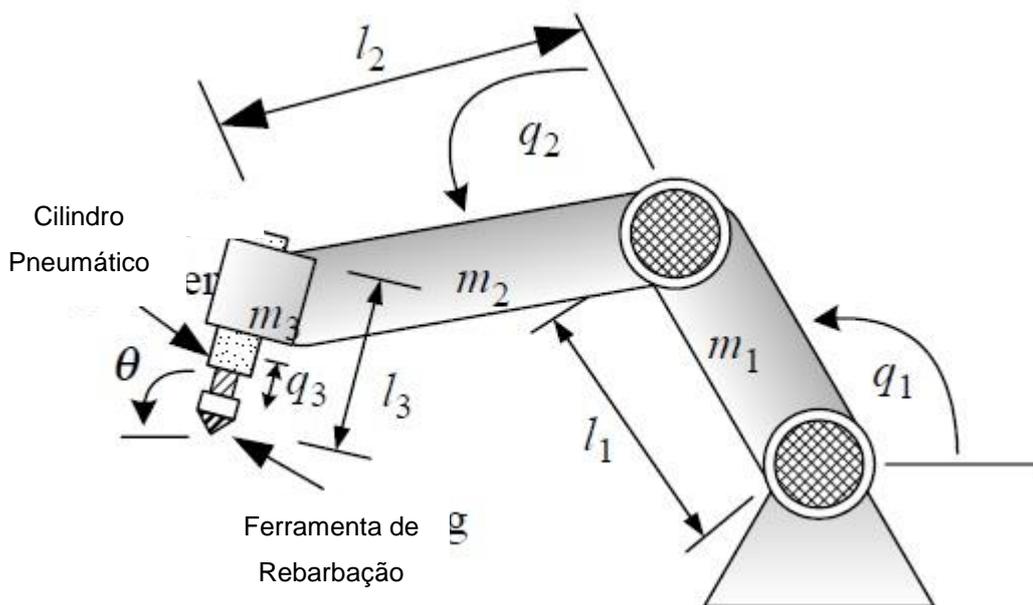


Figura 2.21: Proposta de Changhoon Kim et al [24]

Da mesma forma que o trabalho de Ziliani et al., o trabalho de Pan e Zhang 8, também propõe um método para correção da trajetória do robô por controle de torque.

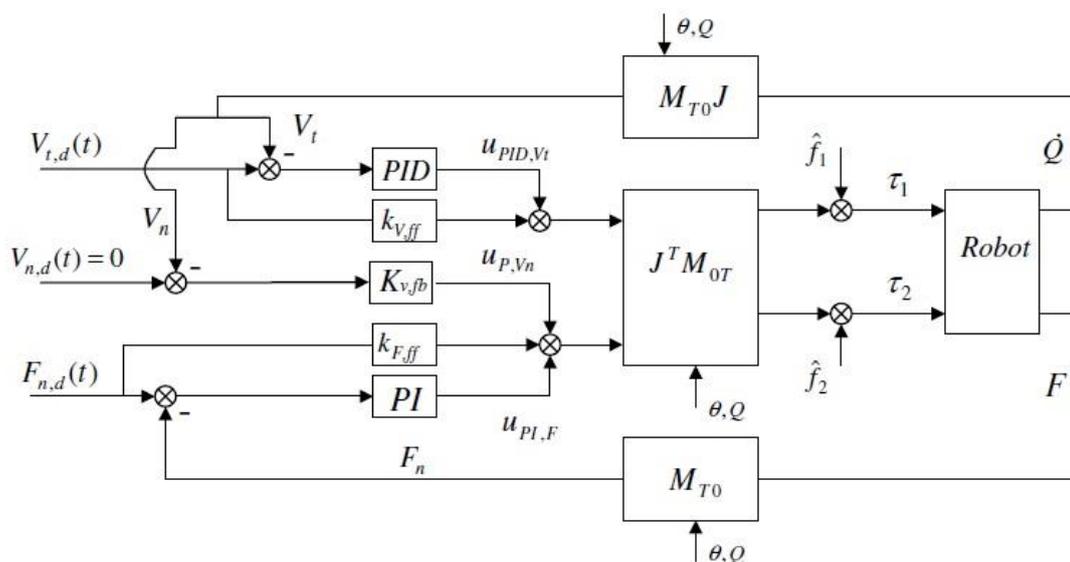


Figura 2.22: Esquema da malha de controle proposta pelo artigo. [40]

De forma similar ao apresentado, os dados da trajetória originais e os dados de sensores de força são parâmetros de entrada de um método que recalcula a trajetória e envia os dados aos servo-motores do robô corrigidos a fim de compensar esses erros e desvios. A Figura 2.23 ilustra o trabalho.

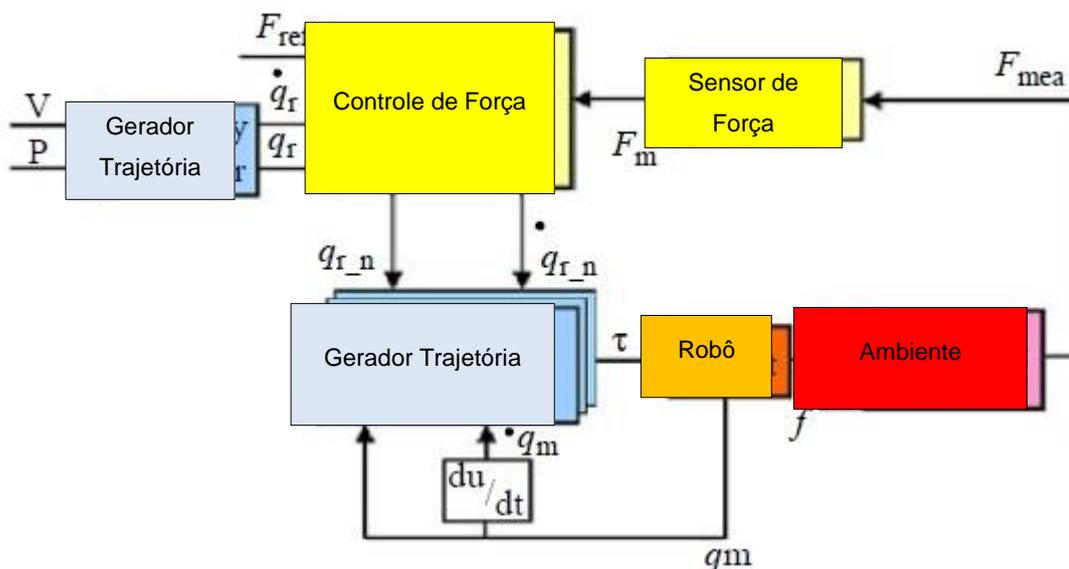


Figura 2.23: Proposta de Pan e Zhang para controle de forças na usinagem [8]

### 2.2.3 Geração de trajetória para robôs

Para a geração de trajetórias para robôs, o trabalho de Nagata et al., [41], descreve o desenvolvimento de um sistema *CAM* para robôs industriais baseado nos controladores dos servos sem usar a linguagem do robô. O trabalho tem como objetivo esse desenvolvimento de sistema *CAM* direto pois afirma que a relação entre *CAD/CAM* para robôs industriais não está tão bem estabelecida quanto para máquinas ferramentas. O artigo relata que em geral, o processador principal *CAD/CAM* calcula o *CL DATA* de acordo com a estratégia de cada máquina, então, um pós processador produz um código *NC* adequado para vários tipos de máquina ferramenta usadas atualmente. Além disso, os sistemas *CAM* para máquinas ferramentas já estão tecnicamente maduros. Por outro lado, os sistemas *CAM* para robôs industriais ainda não estão suficientemente desenvolvidos e padronizados, dessa forma, diversos fabricantes ainda usam seu tipo de linguagem em seus robôs. Outro fator importante é que a configuração de robôs necessita de informações extras como por exemplo correções de trajetórias para evitar colisões e singularidades, que são posições onde o comando não consegue calcular a próxima posição devido a não ter uma única solução matemática para tal

problema. Essa é a razão do porquê robôs necessitam de correções de posicionamentos e orientações após o programa ter sido gerado por métodos *on-line* ou *off-line*. Entretanto, o artigo deixa claro que o processo não está completo, pois ainda há a necessidade de fazer alguns setups iniciais com o *teaching pendant* [41]. A Figura 2.24 ilustra o robô utilizado.



Figura 2.24: Robô de arquitetura aberta usada pelo artigo de Nagata At All [41]

## 2.2.4 Processos de acabamento utilizando robôs

Processos de acabamento usando robôs é uma tecnologia recente, emergiu durante a última década e representa hoje, a menor parcela no uso de robôs. Porém, são evidentes que os benefícios econômicos, qualidade do processo, redução de custos, problemas com a legislação de ergonomia e saúde vem fazendo com que essa tecnologia cresça cada vez mais [1].

Abaixo, segundo [1], algumas das aplicações de robôs industriais para processos de acabamento em geral e um exemplo com o foco em rebarbação utilizando de robô.

Na Figura 2.25 pode se observar o uso de um robô para guiar a peça na máquina de polir. Essa tarefa antes era realizada por um operador. Pelo mais habilidoso que o mesmo fosse, a trajetória e a força exercida sobre a peça em relação a máquina não são sempre a mesma. Além do fato de ser um trabalho repetitivo. Com a aplicação do robô, esse pode realizar inúmeras vezes a mesma tarefa, se não continuamente e de forma repetitiva e com repetibilidade.

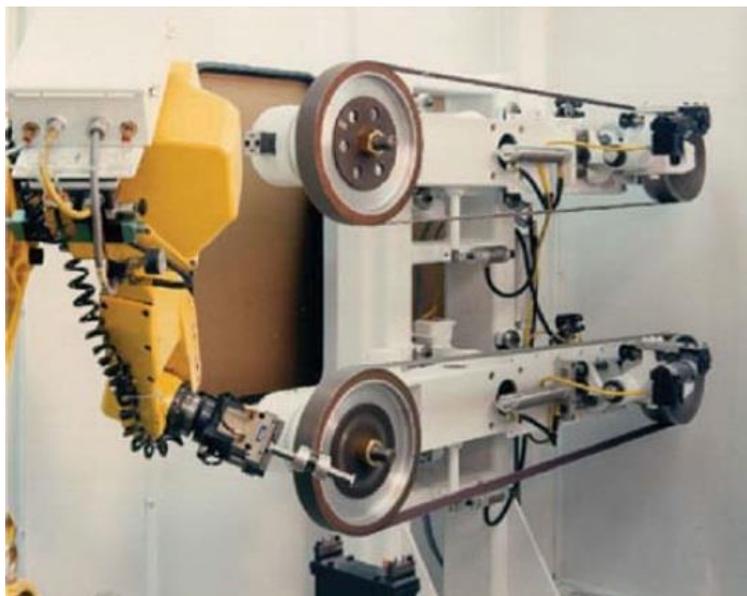


Figura 2.25: Polimento [1]

Da mesma forma, a Figura 2.26 ilustra outra aplicação. Nesse exemplo, a ferramenta está acoplada ao robô, e a peça está em uma mesa rotativa.



Figura 2.26: Polimento [1]

Na Figura 2.27 pode-se observar um exemplo de retificação, onde o robô guia o rebolo retificador em torno da peça.

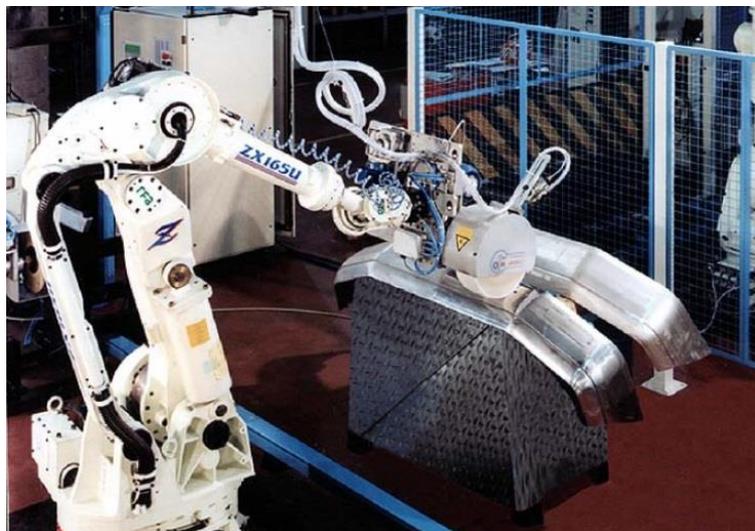


Figura 2.27: Retificação [1]

E como último exemplo, a operação de rebarbação realizada por um robô ilustrado na Figura 2.28. Portanto, se torna claro que robôs podem ser utilizados para operações de acabamento oferecendo inúmeros benefícios. Dentre essas operações, a rebarbação é o exemplo utilizado na realização desse trabalho.



Figura 2.28: Rebarbação [1]

Como justificativa do processo de rebarbação como processo de acabamento, para acabar peças, como exemplo, componentes de motores aeronáuticos são

submetidos a altos níveis de estresse e vibrações durante sua operação. A integridade mecânica desses componentes pode ser comprometida pela presença de rebarbas ou cantos vivos como arestas. Portanto, a remoção de rebarbas e o arredondamento de cantos vivos se faz necessário para melhorar a resistência térmica e mecânica a fadiga. A maioria das operações de rebarbação são efetuadas manualmente por ferramentas rotativas com vibrações excessivas e esforços repetitivos que representam riscos para a saúde e segurança do operador. Além do fato dos possíveis erros que podem danificar peças caras além do ponto de reparo, no sentido de que se o operador cometer um erro de geometria grande ao rebarbar, essa peça pode ser considerada perdida. Algumas operações de rebarbação usam pequenas máquinas CNC ou ainda robôs. Máquinas ferramenta são atrativas pela sua rigidez mecânica e precisão, porém, os custos dessas máquinas são muitas vezes demasiadamente caros para um processo de rebarbação e essas também são inflexíveis devido ao número limitado de eixos e curso para os movimentos. Em contraste, robôs são menos rígidos e precisos, porém, oferecem grande volume de trabalho e maior flexibilidade de movimentos [9].

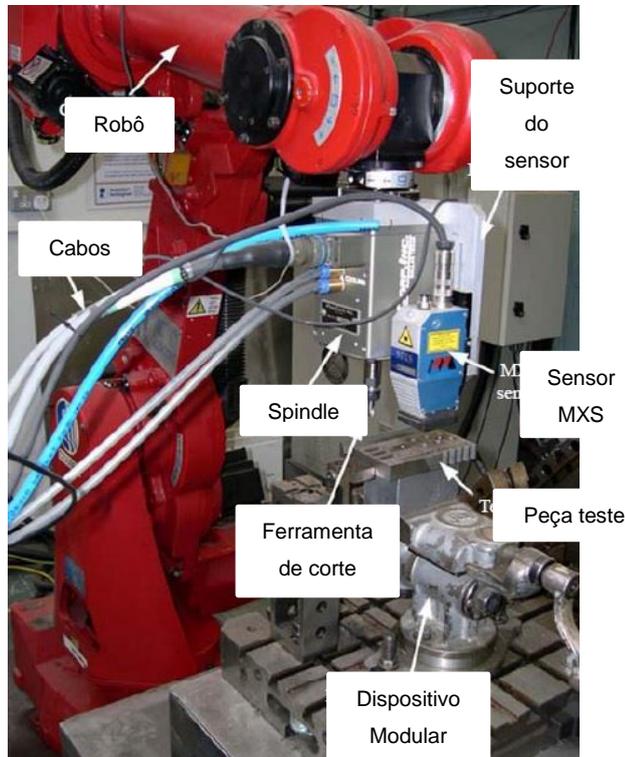


Figura 2.29: Hardware do artigo [9]

A Figura 2.30 ilustra a arquitetura do controle descrito acima, utilizado pelo trabalho.

A rebarbação por robôs industriais tem sido de maneira geral, realizada através da programação por *teaching*. A desvantagem desse método é que se assume que a peça estará sempre nas mesmas condições e portanto, pode ser usado sempre a mesma trajetória para a ferramenta. Isso acarreta no problema que alguma rebarba pode não vir a ser removida ou ainda, a ferramenta pode estar passando em regiões desnecessárias. Ainda mais, para geometrias complexas, é necessário a programação de centenas ou até milhares de pontos para o robô executar a tarefa com precisão. Sendo assim, a programação por *teaching* para peças de geometrias complexas se torna impraticável. Uma alternativa é usar sistemas CAD para que através do modelo da peça, possa auxiliar nessa tarefa. Com isso a programação se torna mais rápida, mas ainda assim, esse modelo não possui nenhuma informação sobre as irregularidades da peça. Isso pode ser obtido através do uso de um sensor que faça a medição das arestas em tempo real [9].

Arquitetura do controle

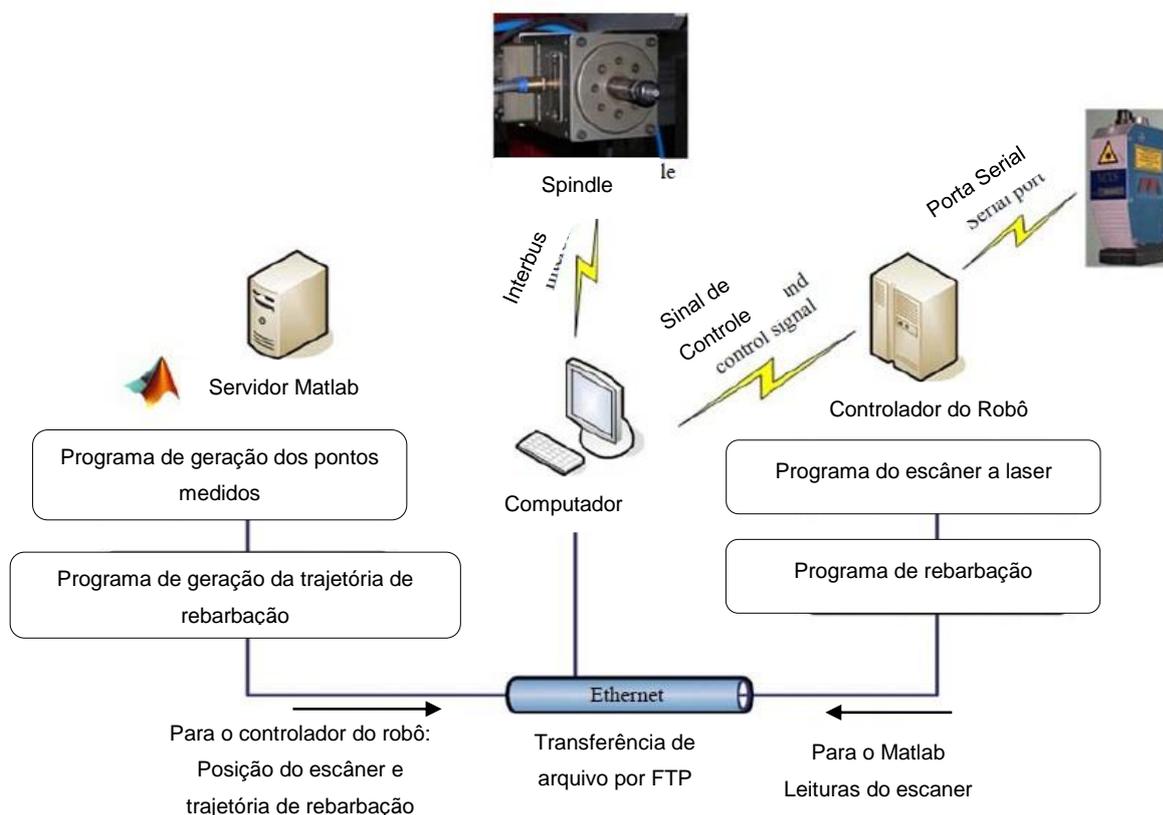


Figura 2.30: Arquitetura da metodologia do artigo [9]

O método proposto por Jayaweera et al., [9], consiste em por exemplo, medir uma série de pontos na área a ser rebarbada. Então, essa geometria é carregada no matlab junto a um programa já feito que usinaria a geometria em questão. Então, através da metodologia matemática de *best fit*, a diferença entre a trajetória

previamente programada e a atual geometria dará o tamanho e dimensão das rebarbas. Então essa trajetória é adequada a geometria escaneada e enviada ao robô através de protocolo TCP. A Figura 2.29 ilustra os testes do artigo.

### 2.3 Processo de escaneamento

Atualmente, as tecnologias para o escaneamento de objetos 3D vêm crescendo cada vez mais. Suas aplicações estão desde área da medicina, moda, medição de superfícies de moldes, prototipagem rápida, engenharia reversa e a área industrial. As aplicações variam entre controle de qualidade geométrica, avaliação de superfícies, engenharia reversa entre outras [42, 43, 44].

Recentemente, com o desenvolvimento e progresso de áreas como a ótica, microeletrônica, automação, além dos escaners tradicionais, os portáteis tem sido alvo de grande interesse. Eles podem ser combinados com um robô industrial para adquirir a informação 3D através de nuvem de pontos da superfície de objetos a partir de vários ângulos e direções de escaneamento e ainda, diminuir problemas como oclusões ou sombras [44].

Dentre muitas tecnologias para a obtenção de nuvens de pontos, destacam-se: digitalização a laser (*scanning systems*), processo por aquisição de fotos (*photogrammetry*) que também é conhecido por *Multi-View Stereo* (MVS), sistemas de luzes estruturadas (*structured lighting systems*) ou ainda, máquinas de medir por coordenadas (*coordinate-measuring machines*) [18].

Dentre os processos mais conhecidos de escaneamento 3D, pode ser citado o uso de um escâner fixo e o objeto em uma mesa rotativa; CMM (*Coordinate Measuring Machine* – Máquina de Medir por Coordenadas) utilizando um escâner como ferramenta de medição acoplado no lugar do apalpador; *Hand Scan* onde o processo se baseia a um escâner portátil de manipulação através das mãos do operador; processos de escaneamento por robôs, onde escaners são fixados no elemento terminal do robô e esse o conduz em torno do objeto a ser escaneado, dentre outros [43].

Processos de escaneamento óticos são menos precisos que outros processos como medição por máquina de medir por coordenadas, porém, na aquisição de dados, são gerados um número muito grande de pontos por unidade de área e tempo, facilitando assim a aquisição de superfícies e geometrias [42].

Um sistema de medição 3D por laser tem muitas vantagens, dentre elas, destacam-se a não necessidade de toque na peça, a velocidade de aquisição, o

arranjo ótico simples, baixo custo e ainda, a boa estabilidade do sistema. Os sistemas de medição a laser podem ser combinados com um robô industrial para adquirir a informação 3D, ou seja, a nuvem de pontos da superfície de objetos a partir de vários ângulos e direções [44].

Hosseininaveh et al., em seu artigo afirma que o processo de aquisição de dados por fotos (*photogrammetry*) estão cada vez mais disponíveis tanto para usos profissionais quanto para amadores e abrangem uma variedade de aplicações. O trabalho apresenta basicamente dois procedimentos de escaneamentos. Um usando um escâner 3D e outro usando duas câmeras fotográficas montadas de forma próxima uma da outra, denominado de câmeras estéreo. Esse processo consiste basicamente em extrair medições correspondentes comparando as diferenças ou disparidade das imagens. O trabalho mostra que se for gerada uma boa rede de imagens, que é como o artigo denomina uma sequência de imagens. A nuvem de pontos apresentada pelas duas câmeras chega a ser tão boa quanto a do escâner 3D 45. A Figura 2.31 demonstra o processo de geração da imagem 3D a partir de duas imagens 2D e a Figura 2.32 mostra como foram arranjadas as câmeras fisicamente.

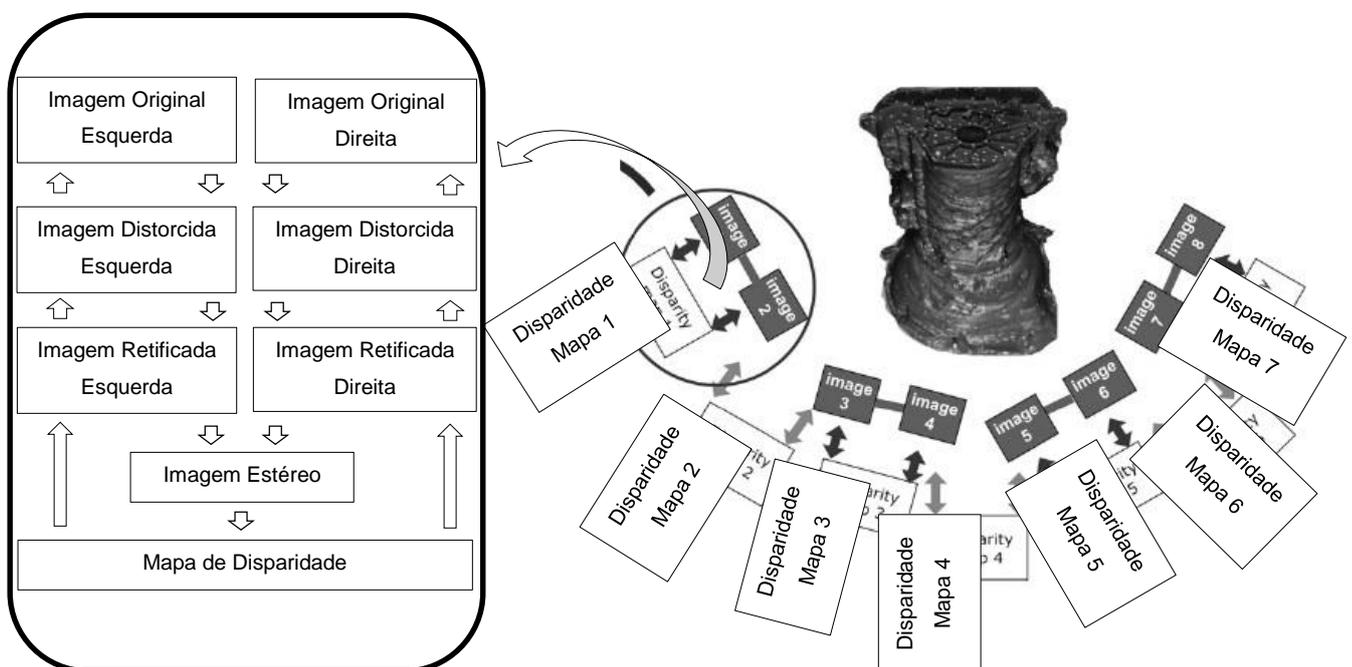


Figura 2.31: Geração de nuvem de pontos através de duas câmeras [45]



Figura 2.32: Câmeras montadas [45]

Dentre as tecnologias para o processo de geração de nuvens de pontos, o processo por aquisição de fotos é a tecnologia com menor custo e boa precisão para essa finalidade. Porém, segundo Ahmadabadian et al., [46], para ser um rival à altura do processo de escaneamento 3D por lasers, o problema do fator de escala tem que ser resolvido. Usando uma fusão de equipamentos de imagens poderia eliminar a necessidade de saber a distância dos objetos da câmera. Outro fator que deve ser levado em consideração com relação ao processo de aquisição de fotos é que para se obter uma boa resolução de malha da nuvem, necessita-se de muitas imagens ou fotos. Isso requer alta capacidade de processamento para a reconstrução 3D do objeto. Outro problema é que muitas vezes ocorre áreas ausentes, devido à falta de cobertura de todas as áreas e também, muitas vezes ocorrem de imagens serem redundantes o que aumenta o tempo de processamento desnecessário. No caso do uso do processo por câmeras, esse trabalho propõe um método para melhorar a seleção das imagens a serem processadas [46].

O processo de escaneamento assim como o seu resultado, que é a geração de nuvens de pontos, pode ser usada não somente para gerar um modelo matemático a partir de um modelo físico, no intuito de fazer engenharia reversa, mas também, para inspeções geométricas [47].

Inspeção é um processo de determinar se um produto está dentro das suas especificações ou tolerância geométricas. Esse processo consiste em gerar uma nuvem de pontos do objeto a ser analisado e comparar essa nuvem com o modelo CAD desse produto. Esse processo é denominado de inspeção por nuvem de pontos [18].

O trabalho de López et al., [18], estuda processos estatísticos para a inspeção através de nuvens de pontos. Esse artigo relata que em geral, o processo de comparação do modelo escaneado com o modelo CAD consiste basicamente em calcular a distância entre esses modelos. Entretanto, antes de realizar esse

processo, essas nuvens necessitam de estar alinhadas por processos de comparação de sistemas de coordenadas. O mais comumente utilizado é o ICP – *Iterative closest point* [18].

O processo de medição em si é em geral, realizado manualmente e exige pessoal altamente qualificado. Diz respeito ao operador escolher as posições de medição com relação ao objeto a ser medido [43].

Se o objeto a ser escaneado possuir geometria simples, o número de escaneamentos, ou seja, áreas que o escâner deve percorrer sobre o objeto é menor e a complexidade do posicionamento do mesmo é pequena. Se o objeto a ser escaneado for complexo, a quantidade de escaneamentos é maior, mas ainda, o posicionamento do escâner se torna um desafio a fim de gerar uma nuvem de pontos densa e completa [43].

O mesmo ocorre quando utilizado o processo de escaneamento por fotos, ou seja, aquisição de imagens, o nível de precisão e plenitude desejada depende diretamente da quantidade e posição das imagens. Se o objeto a ser escaneado possui forma simples, a nuvem pode ser obtida a partir de poucas imagens. Se o objeto a ser escaneado for de forma complexa, surge o desafio de identificar as posições para adquirir essas imagens, pois frequentemente algumas regiões ficam vazias após o escaneamento [42].

Qualquer que seja o processo para adquirir a nuvem de pontos, o principal problema é que essas falhas ou vazios nas nuvens muitas vezes só são identificados depois que o processo de escaneamento já está terminado. Metodologias e ferramentas devem ser aplicadas após o processo de escaneamento a fim de amenizar essas falhas, e se torna uma etapa problemática [43].

Dessa forma, encurtar o tempo de digitalização e reduzir os custos do processo principalmente em objetos de geometria complexas tornam interessante e viável o processo de automação desse processo com o intuito de posicionar e traçar as trajetórias do escâner [43].

Em qualquer processo, a regra geral para o processo de escaneamento é baseada na solução dos mesmos problemas, que seguem [43]:

1. Identificação dos melhores posicionamentos do escâner durante as medições.
2. Cálculo das posições das juntas do robô manipulador para o posicionamento seguro do escaner através de engenharia reversa.

### 3. Cálculo de trajetórias a fim de verificar possíveis colisões.

A verdade é que as tecnologias atuais para geração de nuvens de pontos possibilitam a digitalização de formas 3D com alto nível de fidelidade, proporciona uma resolução alta e um alto nível de acuracidade. Entretanto, com a capacidade obter grandes quantidade de dados surge a exigência de extrair informações mais relevantes.

Ou seja, evoluindo a tecnologia de aquisição desses tipos de dados, surge a necessidade de melhor avaliar esses dados, e portanto, a extração de dados numéricos e o processamento desses dados que deverão apoiar as decisões na indústria através de diagnósticos automáticos [17].

#### 2.3.1 Etapas para uso de escaners 3D na obtenção de nuvens de pontos a partir de modelos físicos.

O trabalho de Boppana et al [47], retrata o uso da digitalização 3D para o uso na engenharia reversa. A digitalização 3D, que é processo de escaneamento da peça física, é umas das etapas da engenharia reversa.

Tanto o artigo de Boppana et al. como o artigo de Larsson e Kjellander [48] sugerem uma sequência de etapas para se conseguir uma nuvem de pontos com qualidade a partir de um objeto físico. Essas podem ser ilustradas na Figura 2.33.

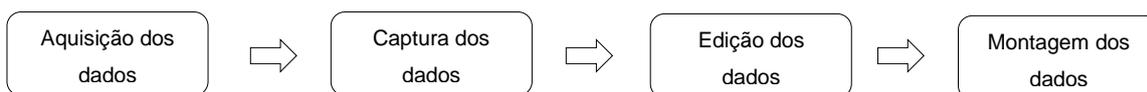


Figura 2.33: Fluxograma da aquisição de dados [47]

As etapas apresentadas na Figura 2.33 são descritas a seguir:

#### **Aquisição de dados**

Nessa etapa, são adquiridos os dados a partir do objeto físico. Normalmente são gerados pontos sem necessidade de estarem em ordem para serem importados em um sistema próprio para o tratamento desses pontos. Basicamente as fases são:

- Fixação do objeto a ser medido
- Digitalização 3D
- Coleta dos dados.

Dentre essas três fases, a digitalização é considerada crucial. Uma série de fatores interferem na incerteza da digitalização, como a orientação, comprimento da trajetória, profundidade da medição, resolução e potência do laser. O trabalho de Bopanna et al., 47, afirma que uma escolha adequada desses parâmetros se faz crucial para uma boa precisão da digitalização.

### **Captura dos dados**

Essa etapa envolve a obtenção de uma nuvem de pontos a partir dos dados obtidos na fase anterior. Como esses dados podem ser originados de um ou mais escaners, podem estar desordenados, sobrepostos, ou seja, o escâner passou várias vezes pela mesma área, entre outros. As operações dessa fase são basicamente registro, limpeza, redução de ruído e triangularização [47].

### **Edição dos dados**

Nessa etapa, o modelo físico é editado rigorosamente para alcançar a representação mais próxima possível do objeto original. Os objetivos dessa fase são alcançar superfícies lisas e limpas, preencher buracos e melhorar a nitidez de arestas [47].

### **Montagem dos dados**

Nessa etapa, envolve a criação de modelos *B-splines*, detecção e extração de contornos e *features*, superfícies planas, arestas, entre outros [47].

## **2.3.2 Métodos de Calibração para escâner a laser 3D**

A calibração de um digitalizador tridimensional é de grande importância quando se busca fatores como boa qualidade, acuracidade, repetibilidade entre outros [49].

Niola et al. [49] propõem um método de calibração para sistemas de escaneamento a laser especificamente para robôs. O método do sistema de aquisição se baseia em um emissor de laser e uma *webcam* fixada em posições relativas. Além disso, uma lente cilíndrica permite ainda a projeção de um plano de luz de laser. Um filtro é utilizado de forma a dividir essa luz. Para o sistema de calibração, foi utilizado um micrometro que se move sobre alvos com dimensões conhecidas. O método é baseado entre medir a relação geométrica entre as coordenadas 3D do laser dividido e as coordenadas digital no plano formado

também pelo laser. O trabalho afirma que através desse método é possível calibrar todo o sistema de forma eficiente [49].

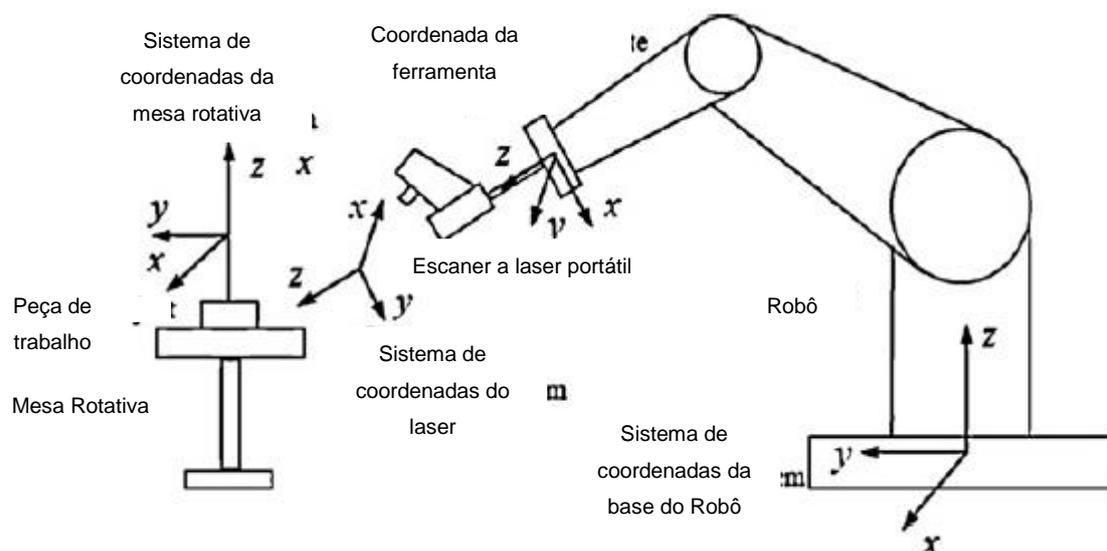


Figura 2.34: Ilustração de como deve ser montado a câmera e a mesa rotativa. [44]



Figura 2.35: Ilustração do escaner portátil fixado ao robô [44]

Visando o problema de profundidade a ser escaneado, ou seja, o quão longe uma determinada região está distante do escâner, devido ao mesmo não conseguir se aproximar mais, e ângulos, o trabalho de Jianfeng et al. [44], propõe um método

de calibração de escâners portáteis a fim de melhorar a precisão do sistema. O artigo sugere a adição de uma plataforma giratória no volume de trabalho do robô a fim de melhorar o acesso do robô ao objeto a ser escaneado [44]. A Figura 2.34 ilustra a proposta do artigo. Já a Figura 2.35 ilustra a montagem do sistema de escâner no robô.

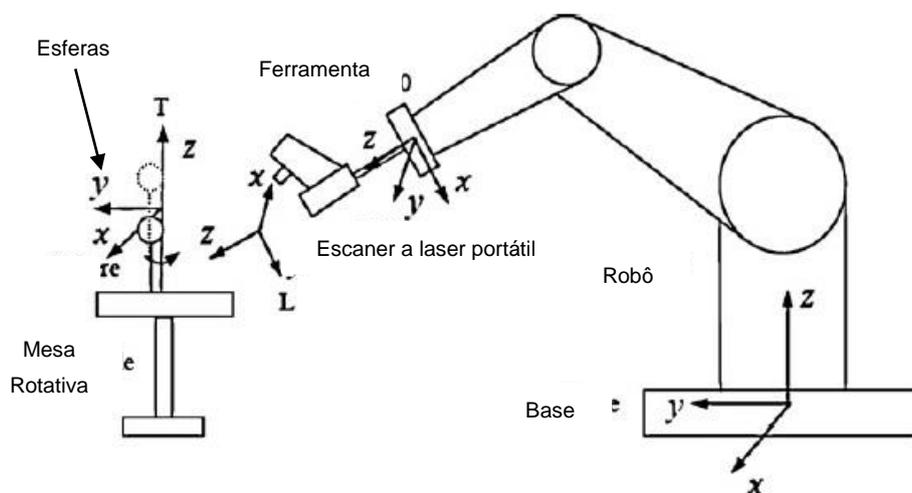


Figura 2.36: Sugestão do trabalho para a calibração da mesa [44]

Para fazer o referenciamento da plataforma giratória, a dificuldade em fazer o alinhamento correto do eixo Z da plataforma, ou mesa indexadora. Para solucionar esse problema, de forma resumida, os autores sugerem a calibração da mesa usando esferas padrões. O laser localiza as esferas padrões e realiza a medição das mesmas. A Figura 2.36 ilustra o processo. Na figura, a mesa com as esferas acopladas na mesma. O procedimento consiste em dois movimentos de translação a fim de determinar o componente de rotação e então, três movimentos de rotação são realizados a fim de determinar o componente de translação para referenciar o escâner em relação ao efetuador do robô. Para o zeramento da mesa em relação ao escâner e principalmente encontrar o eixo de rotação da mesa rotativa, o trabalho sugere alguns movimentos de rotação da mesa rotativa e duas alturas diferentes da posição da esfera em relação a mesa.

## 2.4 Processos de escaneamento automatizados

Os processos de escaneamento são realizados em geral por operadores qualificados, ou seja, manualmente. Quando se deseja automatizar o processo, de forma que uma máquina guie o escâner entorno do objeto a ser medido, em sua maioria são utilizando robôs. O processo consiste em fixar um escâner na

extremidade do robô, possibilitando assim, que o robô faça a trajetória da câmera sobre o objeto. Esse processo se torna viável quando se deseja escanear por mais de uma vez um mesmo objeto ou quando se deseja de alguma forma automatizar até a geração da trajetória que o escâner deverá percorrer. Assim, o programador necessita apenas programar a trajetória uma vez de forma que o resultado seja satisfatório, que então, o robô executará o mesmo processo quantas vezes forem desejadas. Ou ainda, considerando um sistema que possua algoritmos para gerar essa trajetória automaticamente, nem essa fase inicial do operador se faz mais necessária [43]. A seguir são apresentadas algumas das estratégias encontradas na literatura.

Hosseininaveh et al. [42] cujo trabalho tem como objetivo a aquisição automática de nuvens de pontos 3D, descreve em seu artigo uma proposta de sistema para a digitalização 3D onde esse reúne diferentes sistemas e ferramentas em uma plataforma capaz de gerar automaticamente o modelo 3D preciso e completo do objeto a ser escaneado. Esse método consiste na captura de imagens através de uma câmera colocada em um robô de seis graus de liberdade. Essas imagens são então usadas para a reconstrução do modelo 3D usando o método *photogrammetric multi-view stereo*.

O trabalho propõe uma metodologia utilizando de diversos softwares para gerar uma rede de imagens e com elas, simular os lugares para melhor posicionamento da câmera com o objetivo de gerar uma nuvem de pontos mais densa e completa. Para isso, esse trabalho propõe métodos de cinemática reversa para reposicionar o robô e conseqüentemente a câmera.

Transformar uma cena tridimensional, ou seja, com informações 3D, em uma imagem planificada, 2D, é algo trivial em nossas vidas. Um exemplo disso são as câmeras fotográficas. Se fotografa algo espacial e armazena em algo planificado. Porém, o processo inverso, que consiste em transformar imagens bidimensionais em algo tridimensional é algo que exige certo trabalho.

Basicamente a metodologia consiste em: Gerar uma nuvem de pontos bruta do objeto, usando a tecnologia de duas câmeras conforme já explicado no início do capítulo 2.3 e em seguida, armazena essa nuvem em um determinado sistema computacional. Então, um algoritmo específico encontra as regiões falhas nessa nuvem e automaticamente gera uma trajetória para o robô guiar o escâner a fim de fazer uma nova varredura nessas regiões. Então, com os novos dados é gerado uma nuvem de pontos confiável e precisa. A Figura 2.37 mostra o esquema de

posicionamento da câmera e um sistema de braço que avança em dois eixos, e a mesa a qual o objeto a ser escaneado é fixado, com o seu grau de liberdade de rotação [42].

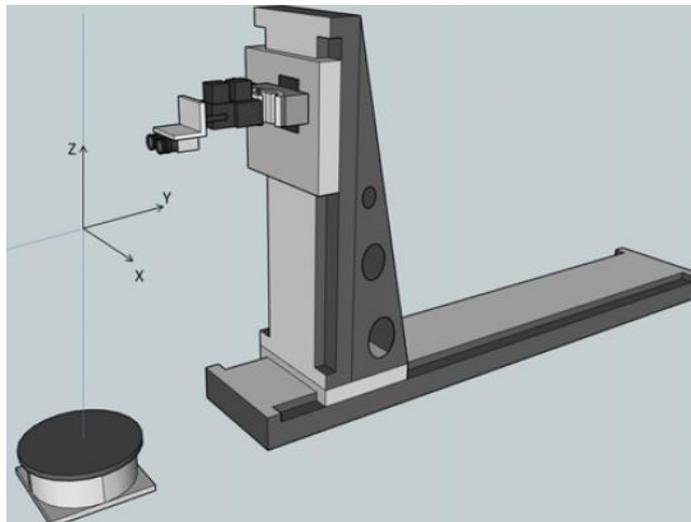


Figura 2.37: Dispositivo para escaneamento do objeto. [42]

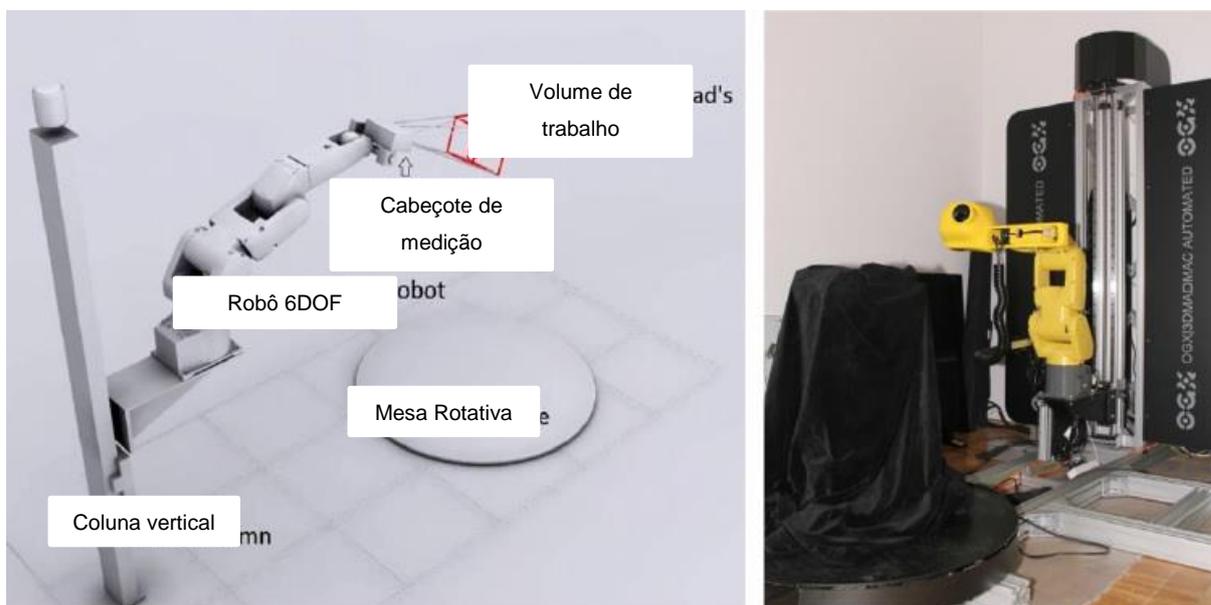


Figura 2.38: A esquerda, modelo do sistema. A direita, modelo físico do processo. [43]

Karaszewski et al., [43], cujo trabalho é posicionamento de scanner através de medição tridimensional automática e livre de colisões através de método *on-line* para aquisição de nuvem de pontos de objetos culturais, tem como objetivo fixar um escâner na frente de um braço robótico, e o objeto a ser escaneado, ser posicionado numa mesa rotativa. A Figura 2.38 demonstra o trabalho proposto pelo artigo. Nessa figura, também pode-se observar a necessidade do ajuste da

distância do escâner em relação ao objeto a fim de manter a área a ser escaneada dentro do volume de trabalho do escâner.

Basicamente o trabalho apresenta um sistema para digitalização 3D de objetos culturais o qual permite um funcionamento totalmente automatizado. Ele identifica as posições mais favoráveis para o escaneamento e através de cinemática inversa, verifica possíveis colisões do braço robótico e ou a câmera com o objeto em questão. O sistema usa de softwares comerciais para realizar os cálculos. O sistema foi aplicado na digitalização de uma coleção de peças do museu Wilanow Palace [43].

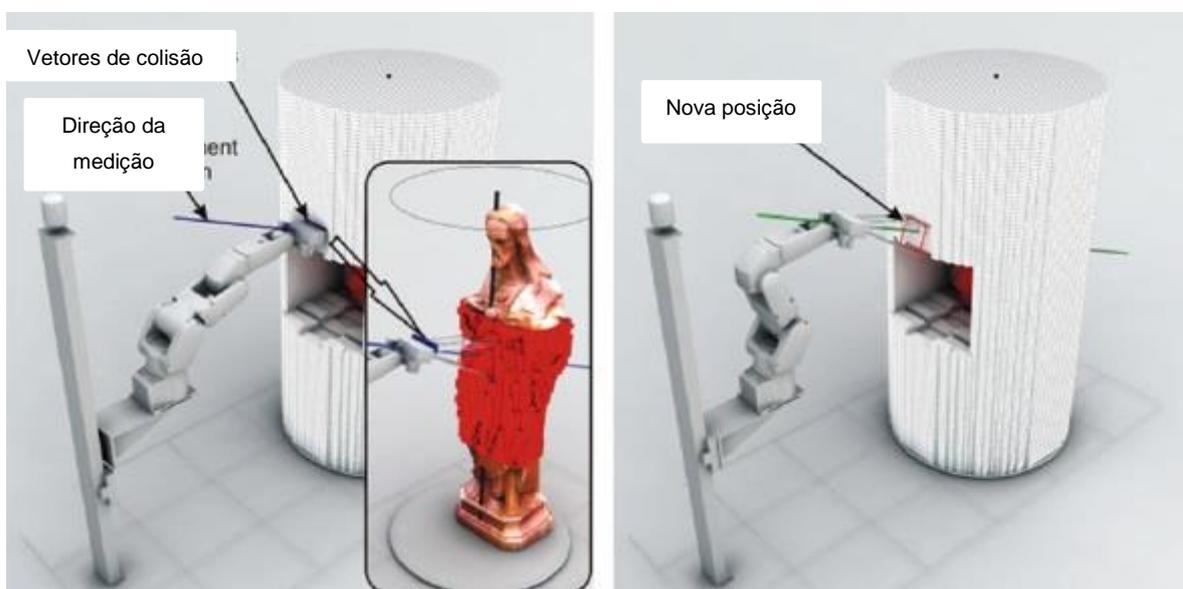


Figura 2.39: Processo de cálculo de posições para o escaner [43]

A Figura 2.39 demonstra como se baseia o processo de posicionamento do objeto no ambiente de trabalho e a metodologia para escanear e detectar colisões. O sistema se baseia na varredura de um cilindro, isso é, o robô apenas faz uma trajetória linear com o escâner enquanto a peça gira em torno de seu eixo através da mesa rotativa. O próprio dado do processo de escaneamento é usado para saber a distância do objeto do escâner e assim, reajustar a distância do escâner ao objeto. Dessa forma o processo se caracteriza como *on-line*.

Já a Figura 2.40 demonstra o processo sendo aplicado fisicamente. Na imagem pode-se reparar que o robô é responsável pela trajetória do escâner e o objeto fica em cima da mesa rotativa.



Figura 2.40: Processo físico escaneando objeto real [43]

O trabalho de Vucina et al., [17], tem como objetivo propor um sistema de avaliação de superfícies 3D baseado na extração de *features*. Basicamente, a nuvem de pontos obtida por escaneamento 3D é comparada com o modelo CAD com a finalidade de detectar desvios de forma. Essa metodologia proposta é demonstrada na forma de detecção e reconhecimento de danos por impacto e cavidades, frestas ou trincas por fadiga ou desgaste por deterioração das superfícies da lamina de turbinas eólicas [17].

A Figura 2.41 ilustra o uso da metodologia do trabalho.

Um sistema de medição ou escaneamento 3D totalmente automatizado não é uma tarefa simples. Precisa do conhecimento detalhado do objeto a ser medido. Isso pode ser facilmente obtido quando o objeto é uma peça mecânica com modelo CAD 3D do projeto. Porém, há casos onde não se possui o modelo CAD ou quando por exemplo, o objeto é um artefato único como uma escultura. A descrição da forma do objeto se faz necessária, segundo Karaszewski et al., [50], para se criar uma lista de posições em que o escâner deverá fazer as leituras para poder escanear completamente o objeto. Para um sistema de leitura de objetos singulares, a lista de medições deve ser atualizada durante o processo de medição. Para um processo automatizado, o sistema de digitalização deve ser

posicionado mecanicamente e a transição entre as posições do escâner devem estar livres de colisões. Esse trabalho tem como objetivo ajudar a identificar as melhores posições para o escâner e através de softwares flexíveis de cinemática inversa, verificar possíveis colisões [50].

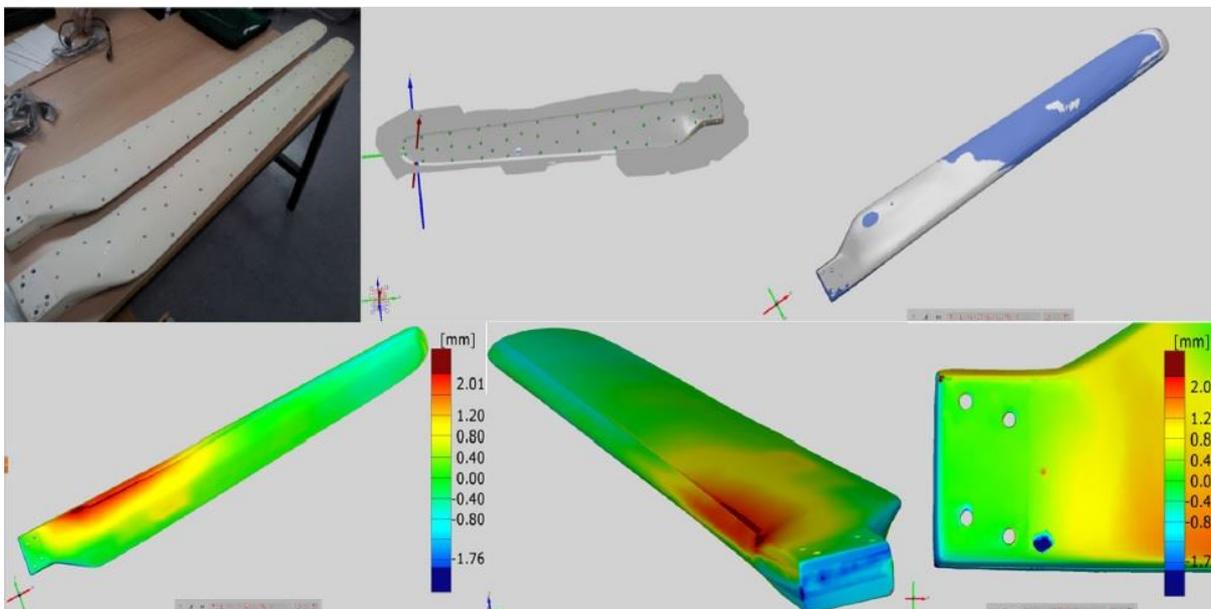


Figura 2.41: Análise de uma lamina de turbina eólica [17]

O trabalho de Larsson e Kjellander [48] utilizam de um sistema CAD para o planejamento da trajetória do robô que guiará o escâner. Através de uma técnica desenvolvida por eles, eles criam pontos para depois traçar curvas por esses pontos que essas servirão de trajetória para o robô. A ideia é, usando o próprio modelo CAD da peça a ser medida, gerar a trajetória. Basicamente, através de uma suavização das faces triangulares da nuvem de pontos que fazem uma intersecção com planos definidos criam curvas para referencias de trajetórias. O modelo proposto pelo artigo consiste em definir pontos no ar (*Air Points*) seguindo uma série de regras definidas no trabalho 48 e basicamente, analisando se a trajetória de transição de um desses pontos para o próximo não ocorrerá colisão do escâner com a peça a ser escaneada [48]. A Figura 2.42 ilustra basicamente a metodologia. Os vetores e sistemas apresentados na figura são melhores discutidos no trabalho, mas representam a forma de localizar a posição do escâner em relação a peça a ser escaneada.

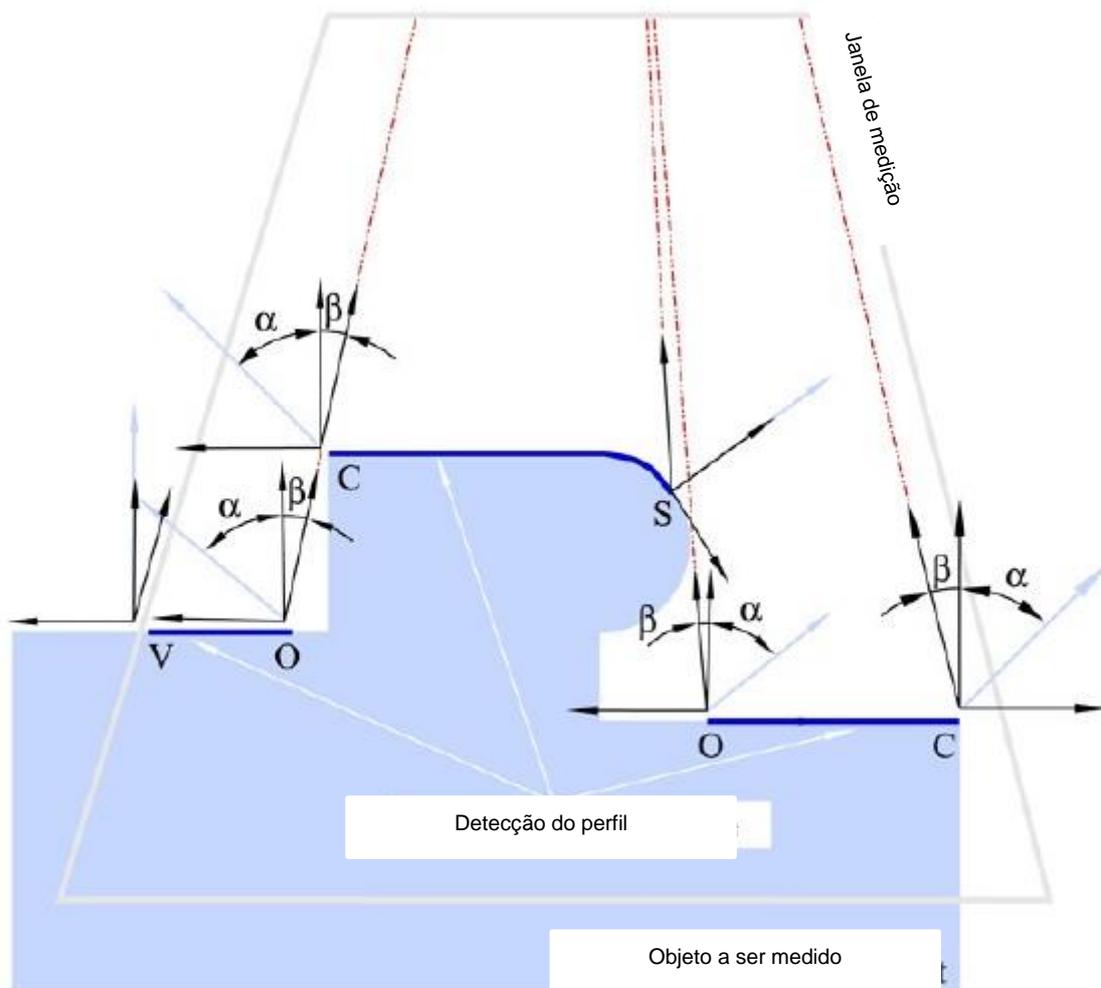


Figura 2.42: Ilustração do método de Larsson e Kjellander para geração automática da trajetória do scanner através de sistemas CAD. [48]

### Definição da mesa giratória

Uma das características positivas no uso de robôs industriais é que seu volume de trabalho é maior do que máquinas convencionais. Porém, quando uma peça de tamanho elevado é colocada a frente do robô para trabalho, esse pode vir a alcançar o lado de trás da peça com dificuldades ou não conseguir alcançar. Vizando solucionar esse problema, os trabalhos de Larsson e Kjellander [48] e o trabalho de Karaszewski et al [43] propõem uma mesa giratória para a realização do processo de escaneamento através de robôs. O principal objetivo dessa mesa é melhorar o alcance do robô que segura o escâner evitando colisões e aumentando seu volume de trabalho.

### **Definição do zero do scanner**

Um dos problemas apresentados nos artigos estudados é a questão do posicionamento do zeramento do sistema de coordenadas do escâner. O trabalho de Larsson e Kjellander [48], relatam que o sistema de coordenadas do objeto a ser escaneado está localizado no centro da mesa giratória.

### **Definição do volume de trabalho**

O trabalho de Larsson e Kjellander 48 descreve o volume de trabalho como sendo um volume que começa na base da mesa giratória e se estende no sentido vertical a mesma por uma distância definida pelas capacidades do *hardware*. Assim também, o raio é definido pela capacidade do *hardware*. Portanto, as limitações do *hardware* que na maioria dos trabalhos é um robô e a base da mesa giratória define o volume de trabalho do sistema de escaneamento [48].

## **2.5 Fundamentos Matemáticos**

O trabalho baseia-se na proposta de um método que não utilize sistemas computacionais já existentes ou minimize esse uso, reduzindo assim a necessidade de programadores, ou pessoal qualificado para tal tarefa. Isso além de reduzir custos, diminui o tempo de programação das tarefas e ainda, reduz erros no processo. Sendo assim, segue uma revisão dos itens considerados mais significativos para o entendimento da proposta dessa tese.

Para o processo de alinhamento, há vários métodos são descritos na literatura, dentre eles destacam-se o algoritmo proposto por Horn, de 1986 e o método conhecido como ICP (*Iterative Closest Point*) ou ponto iterativo mais próximo. Alguns dos artigos apresentados no capítulo 2 sugerem o uso da técnica ICP. Para tal técnica, as malhas devem estar previamente alinhadas. Para isso, método proposto por Horn realiza tal tarefa através do uso de matrizes e seus autovalores e autovetores. Mas ainda, esse método necessita de pontos semelhantes nas malhas. O método proposto por Johnson resolve tal situação. Dessa forma, segue uma revisão sobre esses itens sequenciados com sua necessidade sequencial de execução.

A Figura 2.43 ilustra um processo de alinhamento.

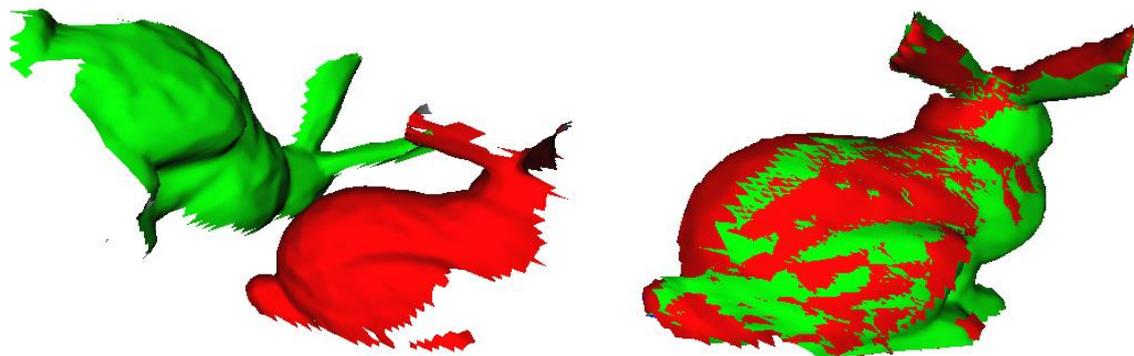


Figura 2.43: Alinhamento de Imagens 3D [51]

### 2.5.1 SpinImages

O método de Horn necessita de pontos semelhantes em ambas as malhas a serem alinhadas. Dessa forma, se faz necessário um procedimento matemático que possa comparar pontos numa das malhas com todos os pontos da outra malha e encontrar o mais correlativo.

O trabalho de Johnson, [51], descreve um método para tal tarefa. Mais informações sobre a técnica podem ser encontradas no trabalho de Johnson ou na dissertação de mestrado de Oliveira, [52].

Porém, a técnica basicamente consiste em um método para reconhecimento de pares de pontos em duas malhas distintas, conforme ilustrado na Figura 2.44.

De forma sucinta, o método consiste em através de equações aplicadas ponto a ponto da malha, que correlacionam todos os pontos de uma mesma malha a um ponto específico dessa mesma malha, gerar dois tipos de imagem. Um, que é um gráfico de pontos. E outro, que através de tons de cinza, demonstra o quão denso, ou seja, o quão mais ou menos pontos uma mesma área na figura tem de pontos quando comparado a outras áreas nessa mesma figura. As figuras com pontos em vermelho são chamadas de *SpinMaps* e a figura com quadriculado em tons de cinza são chamados de *SpinImages*.

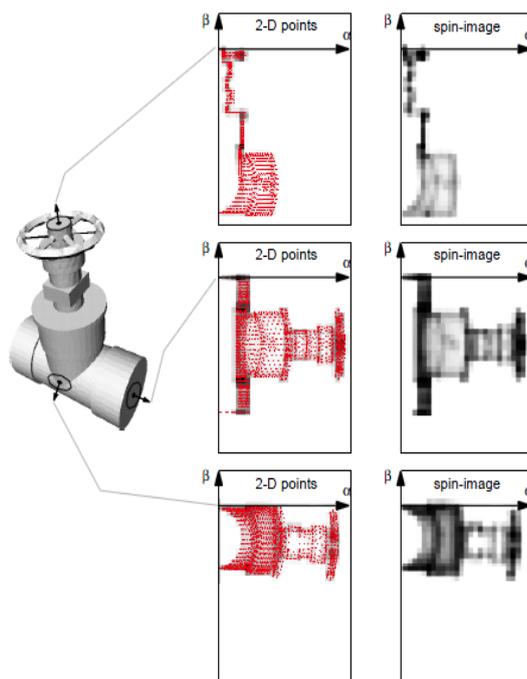


Figura 2.44: SpinMaps e SpinImages [51]

### SpinMaps:

Para a geração dos SpinMaps, a transformação abaixo deve ser aplicada:

$$S_0: R^3 \rightarrow R^2$$

Nada mais é do que transformar as coordenadas dos pontos que estão em 3 dimensões, X, Y e Z em um sistema de coordenadas de duas dimensões, alfa e beta.

Para essa transformação, Johnson propõe o equacionamento:

$$S_0(x) \rightarrow (\alpha, \beta) = \left( \sqrt{\|x - p\|^2 - (n \cdot (x - p))^2}, n \cdot (x - p) \right) \quad \text{Equação 1}$$

Sendo:

$S_0$ : SpinMap correspondente ao ponto orientado p.

X: O ponto em que se pretende projetar no SpinMap

$\alpha$ : Distância perpendicular ao vetor normal n

$\beta$ : Distância perpendicular ao plano P

p: Ponto de referência do SpinMap

n: Vetor normal ao ponto P

A Equação 1 descreve uma planificação dos pontos. É como se imaginariamente, fosse rotacionado um plano em torno do vetor normal a superfície do ponto em questão, e cada um dos demais pontos deixasse uma marca nesse plano. A Figura 2.45 ilustra o equacionamento graficamente.

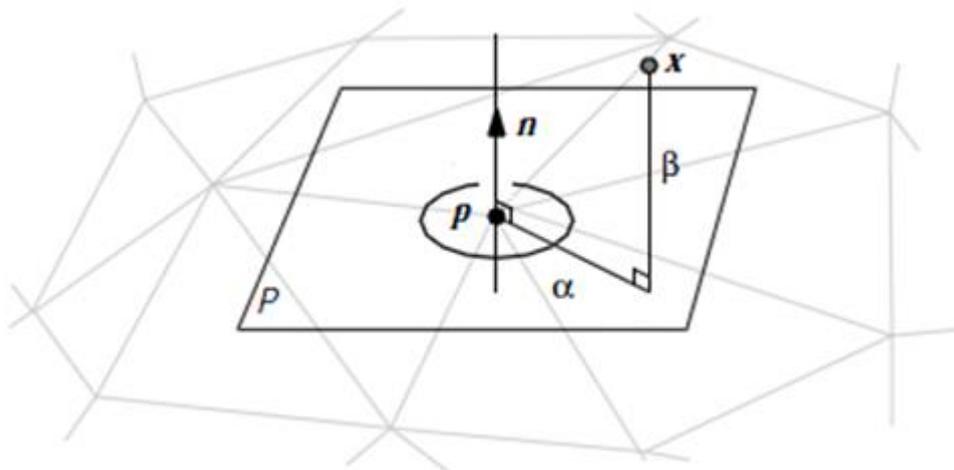


Figura 2.45: SpinMaps [51]

SpinImagens:

Para a transformação seguinte, dos SpinMaps em SpinImagens, deve ser aplicado o método prontos na Figura 2.46.

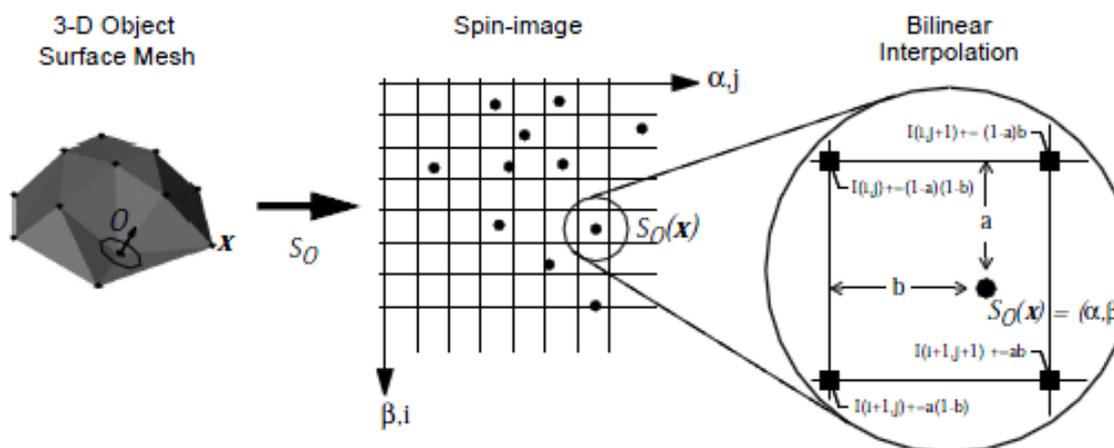


Figura 2.46: SpinImages [51]

As equações seguintes sevem para determinar em qual pixel cada ponto pertence:

$$i = \left\lfloor \frac{\frac{w}{2} - \beta}{b} \right\rfloor$$

$$j = \left\lfloor \frac{\alpha}{b} \right\rfloor$$

Sendo:

w: Tamanho da imagem.

b: Tamanho do pixel

Correlação entre as SpinImages

A equação que realiza essa correlação é representada pela Equação 2:

$$R(P, Q) = \frac{N \cdot \sum p_i \cdot q_i - \sum p_i \cdot \sum q_i}{\sqrt{(N \cdot \sum p_i^2 - (\sum p_i)^2) \cdot (N \cdot \sum q_i^2 - (\sum q_i)^2)}} \quad \begin{array}{l} \text{Equação} \\ 2 \end{array}$$

Em que o coeficiente R(P,Q), que pode variar entre -1 e 1, onde -1 é menos correlativo e 1 é mais correlativo. Dessa forma, através desse coeficiente se encontra os pontos similares em cada malha independentemente da coordenada de cada um desses.

### 2.5.2 Algoritmo de Horn

Berthold K. P. Horn, em 1986 publicou um trabalho chamado *Closed-form solution of absolute orientation using unit quaternions*, que é uma metodologia de alinhamento de imagens através de cálculos matriciais e rotação por quatérnios

A vantagem desse método é que as malhas de pontos podem estar em qualquer posição ou orientação inicial. A desvantagem está em que sua precisão depende das coordenadas dos pontos que forem fornecidos para os cálculos serem os mais correlativos possível, ou seja, Horn ressalta que a propriedade de simetria é importante para o sucesso desse método [52, 53].

Descrição do Método

Dados os pontos como mostra a Figura 2.47, com diferentes coordenadas cartesianas, c e s, a solução dessa transformação é denominada como orientação absoluta [52, 53].

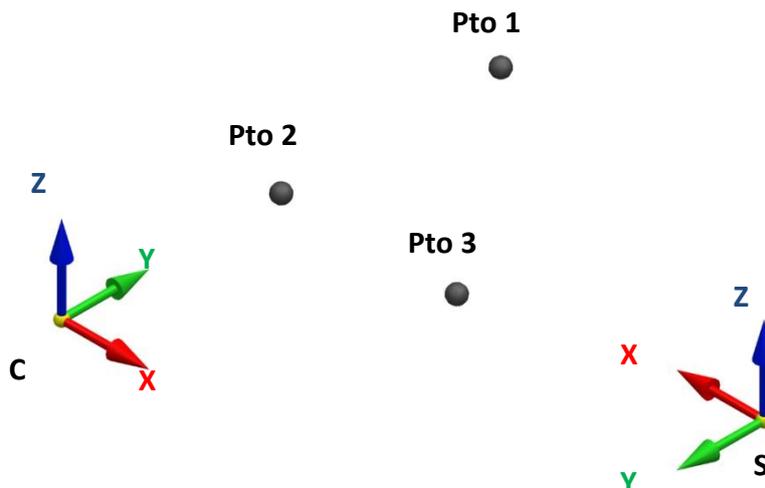


Figura 2.47: Coordenadas dos pontos mensuradas em dois diferentes sistemas de coordenadas [52]

Uma descrição melhor da metodologia pode ser também encontrada na dissertação de mestrado de Oliveira, [52]. Basicamente a transformada entre dois sistemas de coordenadas é o resultado de um movimento decomposto em uma rotação e uma translação. A translação possui três graus de liberdade, a rotação mais três e o fator de escala, que no caso desse trabalho não será usado, mas se faz necessário para os cálculos, mais um grau de liberdade. Portanto, tem-se sete variáveis desconhecidas e se faz necessário então, três pontos no mínimo de cada malha para fornecer nove restrições, três de cada coordenada.

### Translação

Para o cálculo da translação, deve-se calcular os centroides de cada uma das nuvens de pontos. Então deve-se encontrar a diferença entre essas coordenadas. Essa diferença será a melhor translação.

Para encontrar os centróides, a equação a seguir descreve a solução do problema, onde essa consiste basicamente em somar as coordenadas dos pontos, e dividir o resultado pelo número de pontos da malha 52 53.

$$c_k = \frac{1}{n} \sum_{i=1}^n p_{k,i}$$

Sendo:

$c_k$ : é a coordenada X, Y ou Z do centróide

$n$ : o número de pontos da malha

$p_{k,i}$ : O ponto onde  $k$  representa  $x$ ,  $y$  ou  $z$ , e  $i$  o ponto a ser calculado.

### Rotação

Horn mostra em seu trabalho que para tornar o problema da rotação menos complexo, deve-se calcular os centróides das malhas, e então recalculam as coordenadas dos pontos a partir desses centróides. Ou seja, como se o sistema de coordenada de cada malha fossem em relação a esses centroides.

Portanto, para determinar essas novas coordenadas, basta subtrair a coordenada de cada ponto de cada malha de seu devido centroide conforme a equação abaixo:

$$P'_{k,i} = P_{k,i} - c_k$$

Em seguida, deve-se encontrar a componente de rotação que resolva a equação:

$$Q_i = R(P_i)$$

Horn descreve que ao expressar esta matriz de rotação como um quatérnio, pode-se reduzir o problema a determinar o autovetor associado ao maior autovalor da matriz simétrica 4x4.

Para isso, primeiro calcula-se a matriz  $M$ , cujo elemento  $(i, j)$  é a soma dos produtos das coordenadas das duas malhas,  $c$  e  $s$ .

$$M = \begin{bmatrix} S_{xx} & S_{xy} & S_{xz} \\ S_{yx} & S_{yy} & S_{yz} \\ S_{zx} & S_{zy} & S_{zz} \end{bmatrix}$$

Cada termo deve ser calculado como na equação abaixo:

$$S_{kl} = \frac{1}{n} \sum_{i=1}^n P'_{k,i,c} \times P'_{k,i,s}$$

Sendo:

$kl$ : Representam as coordenadas  $x$ ,  $y$  e  $z$ .

$n$ : Número de pontos da malha

$P'_{k,i,c}$ : Ponto referencial na malha  $c$ .

$P'_{k,i,s}$ : Ponto referencial na malha  $s$ .

Em seguida, deve ser calculada a matriz N, através dos elementos calculados na matriz M:

$$N = \begin{bmatrix} (S_{xx} + S_{yy} + S_{zz}) & S_{yz} - S_{zy} & S_{xx} - S_{xz} & S_{xy} - S_{yx} \\ S_{yz} - S_{zy} & (S_{xx} - S_{yy} - S_{zz}) & S_{xy} - S_{yx} & S_{zx} - S_{xz} \\ S_{xx} - S_{xz} & S_{xy} - S_{yx} & (-S_{xx} + S_{yy} - S_{zz}) & S_{yz} - S_{zy} \\ S_{xy} - S_{yx} & S_{zx} - S_{xz} & S_{yz} - S_{zy} & (-S_{xx} - S_{yy} + S_{zz}) \end{bmatrix}$$

O autovetor associado ao maior autovalor dessa matriz N representa o quatérnio de melhor rotação.

A solução do polinômio pode ser resolvida pelo método de Ludovico Ferrari, descrito na dissertação de mestrado do autor dessa tese [52].

Obtidos esses quatro autovalores, encontra-se agora o maior autovalor positivo.

Para encontrar agora o autovetor  $\hat{e}_m$  associado à esse maior autovalor positivo  $\lambda_m$ , desenvolve-se a equação abaixo:

$$[N - \lambda_m I] \hat{e}_m = 0$$

sendo:

N: Matriz N

$\lambda_m$ : Maior autovalor positivo da matriz N

I: Matriz identidade

$\lambda_m \cdot I$ : É a multiplicação da matriz identidade pelo maior autovalor da matriz N

$\hat{e}_m$  : É o autovetor que se deseja encontrar.

O autovetor resultante desses cálculos será na forma:

$$\hat{e}_m = q_0 + iq_x + jq_y + kq_z$$

Calcula-se agora a rotação para cada ponto da malha que se deseja realizar as transformações. A equação que resolve matematicamente a rotação na forma de quatérnios é descrita abaixo. Essa multiplicação não é simples. Deve se obedecer a regra de multiplicação envolvendo números na forma de quatérnios.

$$P' = q \cdot P \cdot q^{-1} \leftrightarrow P' = R(P)$$

### 2.5.3 ICP – Iterative Closest Point

O algoritmo ICP é bastante citado no capítulo sobre processos de escaneamento.

A técnica relata que esse algoritmo serve para o refinamento do alinhamento. Ou seja, que as nuvens de pontos devem estar previamente alinhadas antes de aplicar tal algoritmo. O ICP executa um processo iterativo para refinar a transformação inicial até convergir para um mínimo local. Ao final deste processo, as duas malhas com alinhamento estimado terão um alinhamento mais preciso [52, 54].

A Figura 2.48 ilustra o processo de alinhamento por ICP.

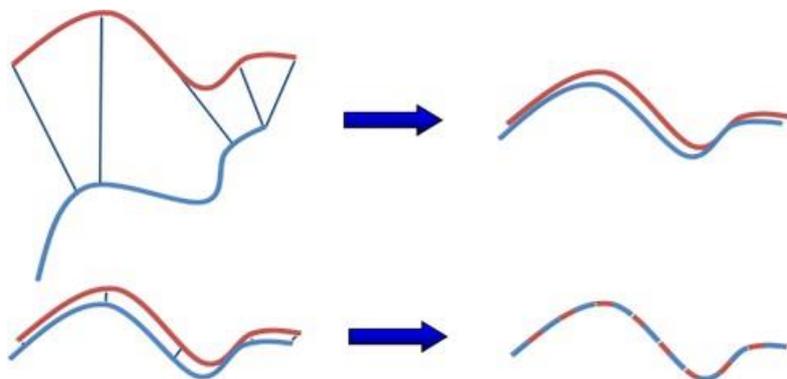


Figura 2.48: ICP interatividade. [54]

#### 2.5.4 Identificação de rebarbas

O método para identificação das regiões e dos volumes de rebarbas foi desenvolvido e implementado no mestrado, [52].

Mais detalhes sobre esse método podem ser encontrados na dissertação [52], porém, segue uma explicação sucinta sobre o método:

O método consiste, de forma sucinta, em depois que as malhas estão alinhadas, sobrepostas, verificar e separar todos os pontos que estiverem distanciadados para fora da nuvem que representa o modelo da peça final de uma distância maior que uma tolerância. Tolerância essa que pode ser determinada com base na tolerância do escaner, e da qualidade superficial do fundido. Ou seja, se algum ponto ou faceta estiver sobressaltado em relação ao STL da peça CAD, essas são rebarbas.

Matematicamente, o método para encontrar as rebarbas consiste em projetar cada ponto da malha escaneada em todas as facetas da peça final e verificar se esses pontos estão dentro de pelo menos uma das facetas da peça CAD e quanto tiverem, se esse ponto está a uma distância maior do que uma tolerância especificada e no sentido do vetor normal. Isso porque, quando contra o sentido do vetor normal, pode ser alguma região que ficou faltando material, e assim, é uma imperfeição na forma de falta de material, e não de sobra como uma rebarba.

Conforme ilustrado a Figura 2.49, muito dificilmente os três pontos do triângulo do modelo escaneado, em verde, terá a sua projeção dentro da área de um triângulo do modelo CAD, em preto. Dessa forma, os três pontos do modelo escaneado podem pertencer ao modelo CAD, em triângulos ou facetas distintas. Pode-se observar também que a projeção do Ponto 1 do modelo escaneado, ponto em azul, pertence a área triangular formada pelos pontos um, dois e três do modelo CAD, em preto, portanto, se a distância normal ao plano não tivesse fora da tolerância especificada, esse ponto para esse triângulo específico não seria uma rebarba. Já a projeção do ponto dois do modelo escaneado, ponto em vermelho, mesmo que a distância normal ao plano não tivesse fora da tolerância especificada, seria para esse triângulo do modelo CAD uma rebarba [52].

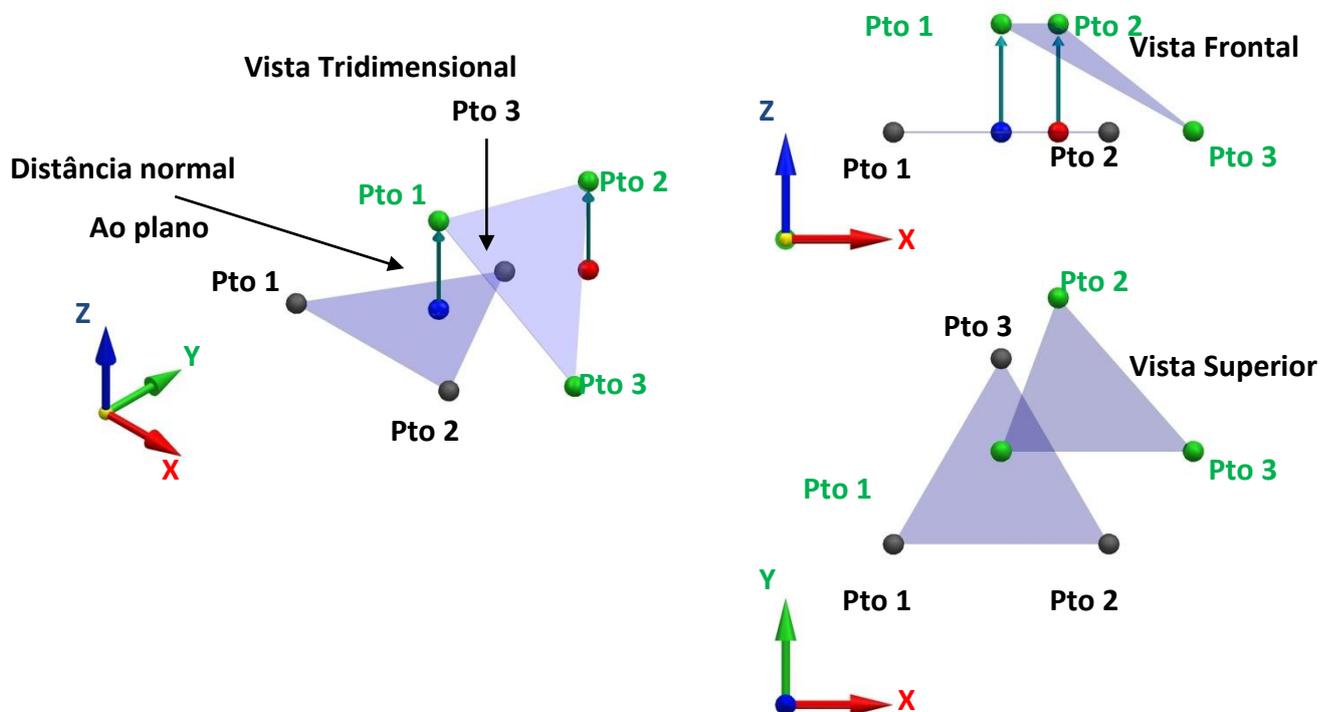


Figura 2.49: Projeção de pontos modelo escaneado em uma faceta do modelo CAD. [52]

A razão pela qual o ponto projetado deve estar dentro da área do triângulo é demonstrado na Figura 2.50. Em relação ao ponto três do modelo escaneado, em verde, sua distância normal ao plano formado pelos pontos do modelo CAD um, dois e três é zero, ou seja, está na tolerância, ou seja, não seria rebarba. Mas quando comparamos ao triângulo formado pelos pontos dois, três e quatro, observamos que esse ponto é uma rebarba, pois, o formato da peça se altera nesses dois triângulos [52].

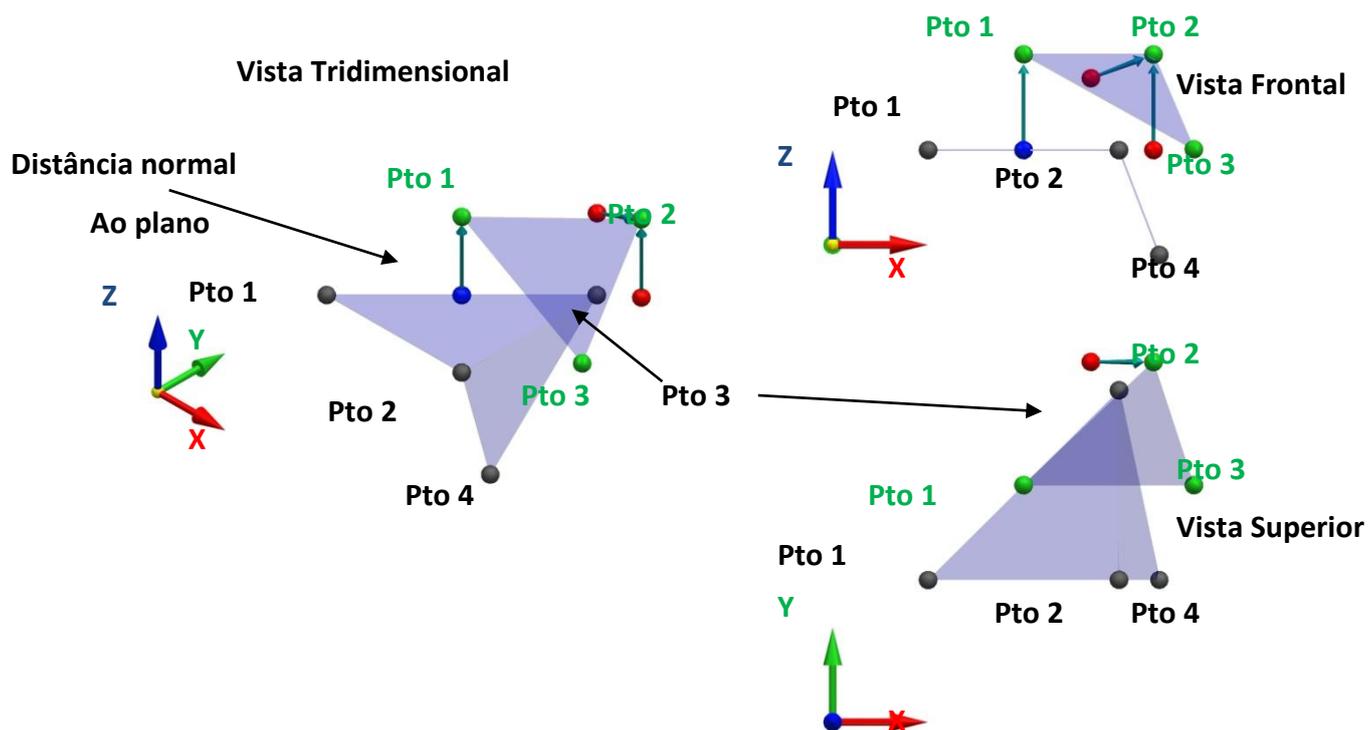


Figura 2.50: Motivo pelo qual o ponto projetado deve estar dentro do triângulo. [52]

Dessa forma, podemos resumir esses casos da seguinte forma:

#### É rebarba se:

- O ponto do modelo escaneado não possuir nenhuma projeção dentro de algum dos triângulos do modelo CAD.
- O ponto do modelo escaneado possuir projeção em algum dos triângulos do modelo CAD, porém, nos que possuir essa projeção, a distância normal ao plano formado pelos pontos que definem esse triângulo é maior que a tolerância especificada.

#### Não é rebarba se:

- O ponto do modelo escaneado possuir em relação a algum dos triângulos do modelo CAD, a distância normal ao plano formado pelos pontos que definem o triângulo em análise.
- Para calcular a tolerância, devem ser levados em consideração dois parâmetros:
  - A tolerância do equipamento usado no processo de escaneamento.

- A exatidão das coordenadas dos pontos fornecidos para o processo de alinhamento.

Para calcular a projeção de um ponto do modelo escaneado a um triângulo específico, deve-se utilizar da Equação 3 **Erro! Fonte de referência não encontrada.**, onde  $n$  é o vetor normal ao plano formado pelos três pontos do triângulo do modelo CAD,  $x$  é o ponto do modelo escaneado e  $p$  um dos três pontos do triângulo do modelo CAD.

$$d = n \cdot (x - p) \quad \text{Equação 3}$$

Deve-se então, subtrair as coordenadas calculadas  $x$ ,  $y$  e  $z$  do ponto do modelo escaneado. Essas novas coordenadas  $x$ ,  $y$  e  $z$  são as coordenadas do ponto projetado no plano. O resultado da equação 4.1 é também, o valor da distância normal ao plano formado pelos três pontos que determinam o triângulo em análise.

Para calcular se esse ponto projetado está ou não dentro do triângulo, podemos fazer uma analogia trigonométrica a Figura 2.51 e a Figura 2.52:

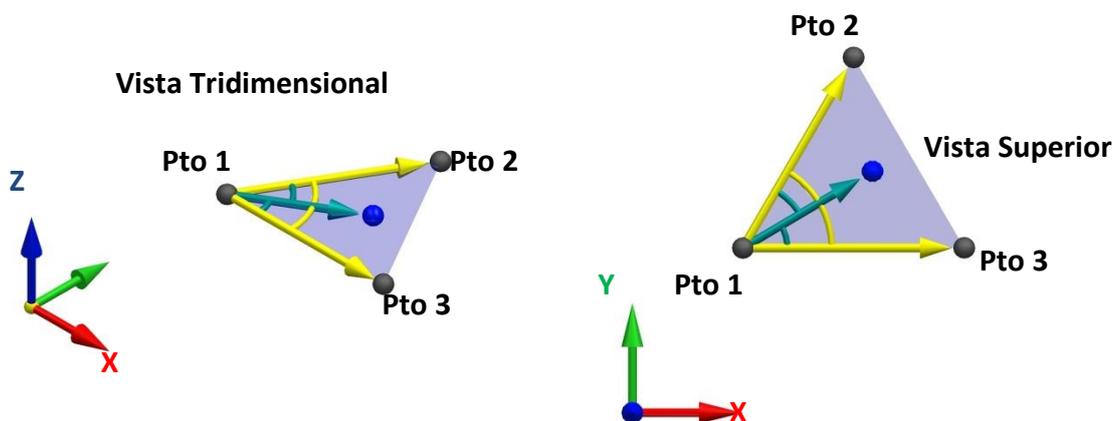


Figura 2.51: Ponto pertencente ao triângulo. [52]

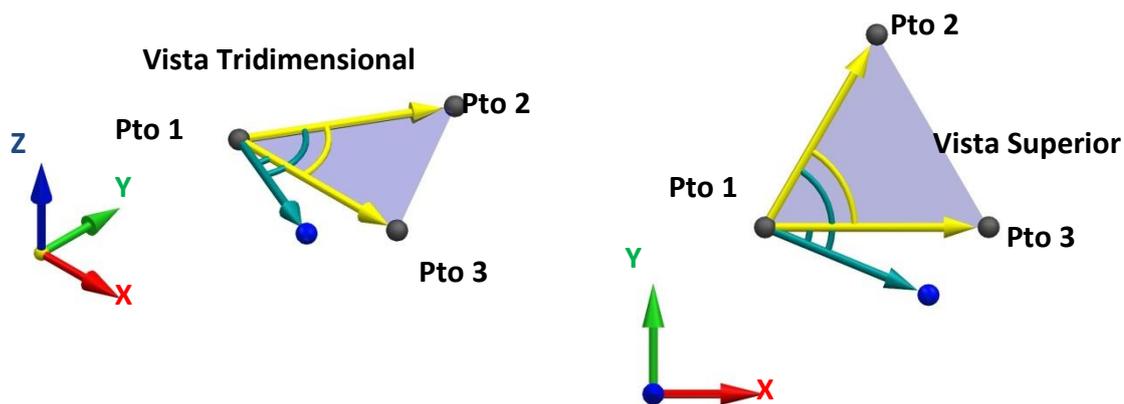


Figura 2.52: Ponto não pertencente ao triângulo. [52]

Ao subtrair o ponto 1 do ponto 2 e o ponto 3 do ponto 1, tem-se os vetores com as mesmas direções e sentido aos vetores representados em amarelo nas figuras. Então, ao subtrair agora o ponto 1 do ponto projetado em azul, tem-se então um terceiro vetor que aponta ponto 1 ao ponto projetado.

Se o ângulo entre o vetor em verde em relação a cada vetor em amarelo for menor que os ângulos entre os vetores amarelos, o ponto pertence ao triângulo olhando por esse vértice do ponto 1. Conforme ilustrado na Figura 2.51. Caso esse ângulo seja maior, esse ponto não pertence a esse triângulo. Conforme Figura 2.52.

Para realizar o cálculo desses ângulos, a Equação 4 deve ser usada, onde A e B são os vetores o qual se pretende calcular o ângulo.

$$\text{arc cos } \theta = \frac{x_a \cdot x_b + y_a \cdot y_b + z_a \cdot z_b}{|\vec{A}| \cdot |\vec{B}|} \quad \text{Equação 4}$$

Esse procedimento deve ser repetido para os três pontos do triângulo em análise, ou seja, os três vértices. Pode-se concluir então que:

#### **Pertence ao triângulo em análise se:**

- Em relação aos três pontos, os ângulos envolvendo o ponto projetado for menor que o ângulo formado pelas arestas do triângulo.

#### **Não pertence ao triângulo em análise se:**

- Em relação a algum dos três pontos, os ângulos envolvendo o ponto projetado for maior que o ângulo formado pelas arestas do triângulo.

Com isso, pode-se então encontrar quais pontos não pertencem ao modelo CAD.

Para encontrar quais facetas não pertencem ao modelo CAD, deve-se separar toda e do modelo escaneado que possuam pelo menos um dos pontos encontrados como rebarba [52].

### 3 Objetivos e Metodologia

Nesse capítulo é apresentado o objetivo principal dessa tese, assim como os objetivos específicos, ou seja, objetivos secundários para se conseguir atingir o objetivo principal, e também, é apresentado a metodologia proposta para conseguir atingir o objetivo.

#### 3.1 Objetivo

Desenvolvimento de um método e sua implementação computacional para identificar as regiões com rebarbas e seu volume em uma peça fundida, injetada ou usinada e gerar automaticamente um programa NC para sua remoção com o auxílio de um cabeçote fresador montado em um robô.

##### 3.1.1 Objetivos Específicos

Para atender o objetivo principal, uma série de objetivos intermediários devem ser alcançados. Dentre eles estão:

- Uso de uma mesa rotativa que se aplique ao processo de escaneamento e que suporte os esforços de corte durante o processo de usinagem.
- Definição de um método para zeramento único do robô, escâner e mesa rotativa. Ou seja, os três processos devem conter o seu zeramento em um ponto comum dentro do volume de trabalho dos mesmos. É importante ressaltar que dentre os trabalhos citados, temos usinagem e escaneamento com robôs. Porém tem-se muito poucos trabalhos citando o uso das duas tecnologias integradas.
- Aplicação dos métodos de Johnson, Horn e ICP para realizar o processo de alinhamento da malha escaneada com o do arquivo CAD proveniente do projeto da peça a ser rebarbada.
- Definição de um método para gerar as trajetórias da ferramenta utilizando um sistema CAM de forma a comprovar que essa geração também pode ser feita de forma automática.
- Pós-Processamento da trajetória da ferramenta para a linguagem do robô. Etapa que se faz necessária pois sistemas CAM geram a trajetória da ferramenta para uma máquina de cinco graus de liberdade, e o robô possui seis. Ou seja, essa etapa deve restringir um dos graus de liberdade para conseguir o código NC para o robô.

## 3.2 Metodologia

Para a realização do objetivo, a seguinte metodologia é proposta e abordada:

- 1 Estudar sobre métodos de programação de robôs, sendo as mais comuns a *on-line* e a *off-line*.
- 2 Estudar trabalhos na área de programação automática de robôs ou correção de trajetórias.
- 3 Estudar o uso de sistemas de escaneamento aplicados a robôs.
- 4 Estudar trabalhos e métodos na área de alinhamento de nuvens de pontos ou malhas.
- 5 Estudar trabalhos e métodos para calcular a diferença entre nuvens de pontos ou malhas.
- 6 Desenvolver um método para o zeramento comum do escâner, robô e mesa rotativa.
- 7 Desenvolver um procedimento matemático para a utilização conjunta dos métodos de Johnson, Horn e ICP para o alinhamento do modelo CAD do produto acabado com o gerado pelo scanner do produto a ser usinado, rebarbado.
- 8 Desenvolver um procedimento matemático para a identificação e separação das geometrias a serem usinadas.
- 9 Desenvolver um procedimento para programar a trajetória do robô com o cabeçote fresador para a remoção das rebarbas.
- 10 Implementar o processo matemático na linguagem JAVA considerando condições de contorno, dessa forma, demonstrar a possibilidade da implementação de todo o processo sem o uso de sistemas comerciais.
- 11 Implementar o processo em um robô industrial utilizando um escâner e um cabeçote fresador ambos comerciais.
- 12 Validar o trabalho através de escaneamentos e usinagem de corpos de prova utilizando um robô industrial.
- 13 Descrever todo o processo na forma de uma tese de doutorado.

A Figura 3.1 ilustra a seqüência das etapas propostas.

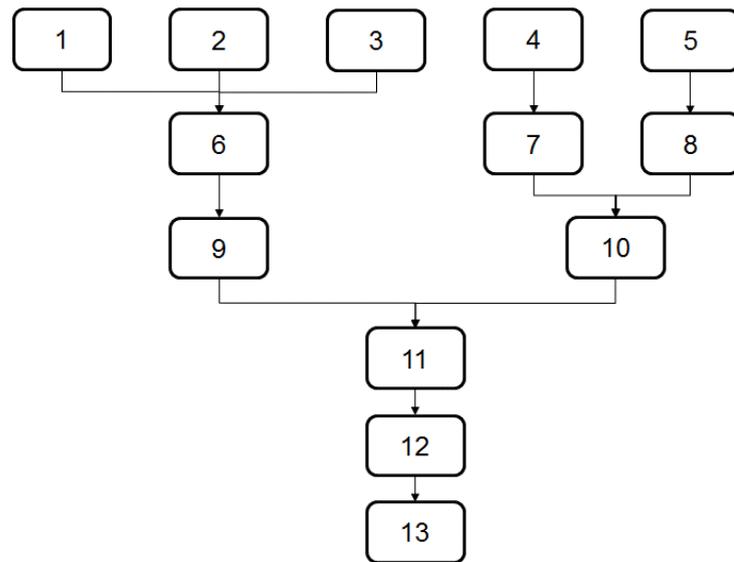


Figura 3.1: Fluxograma da metodologia proposta

## 4 Apresentação geral do conceito e corpos de prova

Nesse capítulo, é apresentado uma visão geral do conceito proposto por essa tese e também, os corpos de prova para a validação do conceito.

### 4.1 Apresentação geral do conceito

A Figura 4.1 demonstra a cadeia como um todo. Em uma visão geral, a figura demonstra a proposta da tese para realizar a programação de um sistema totalmente automático para a realização do processo de escaneamento, assim como todos os cálculos de comparação de geometrias, geração de trajetórias até o pós-processamento para o robô e para a validação, a usinagem dos corpos de prova.



Figura 4.1: Ciclo geral do trabalho

A seguir, cada etapa é descrita de forma mais detalhada.

- **Preparação do sistema:**

Consiste na preparação física do sistema para receber a peça a ser escaneada e rebarbada. Seu objetivo é fixar os dispositivos de escaneamento e usinagem e fazer o setup dos mesmos, conforme ilustrado na Figura 4.2.

**Fixação do escâner:** Deve-se primeiro definir um escâner para o uso no projeto, e então, com o escâner definido, ocorre a fase de projeto e fabricação do dispositivo de fixação no robô. Esse dispositivo tem como objetivo prender o escâner à flange do robô. Nos demais escaneamentos, apenas a montagem no robô quando se fizer necessário, ou seja. O projeto e construção do dispositivo de fixação ocorre apenas uma vez. Com o dispositivo fabricado, toda vez que se trocar a peça a ser rebarbada ou quiser realizar uma nova programação a cada lote de peças por exemplo, basta montar o dispositivo no robô.

**Fixação do cabeçote fresador:** Nessa etapa, também deve ser projetado e fabricado um dispositivo que permita fixar o cabeçote fresador à flange do robô. Também, a parte de projeto e fabricação do dispositivo só ocorre uma vez. Após essa, sempre que se tiver a necessidade do uso do cabeçote fresador, basta montar o dispositivo à flange do robô para o uso do cabeçote.

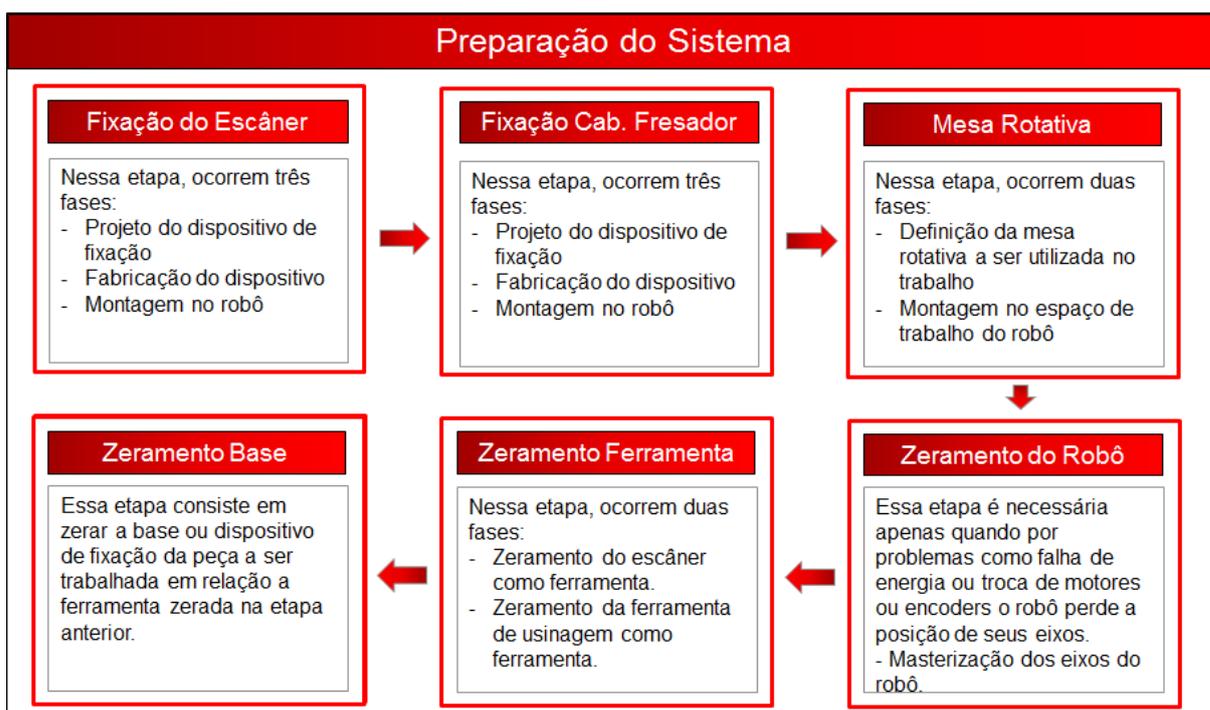


Figura 4.2: Preparação do sistema

**Mesa rotativa:** Uma mesa rotativa ou indexadora acoplada ao volume de trabalho do robô permite uma maior acessibilidade do escâner e até mesmo da ferramenta de usinagem. A diferença do uso de uma mesa rotativa nesse trabalho dos demais apresentados na revisão bibliográfica é que essa deve resistir aos esforços de cortes gerados durante o processo de usinagem.

Zeramento do robô: Para que o robô possa fazer uma trajetória linear por exemplo, esse tem que por meio de equações de cinemática e dinâmica, interpolar seus eixos. Para isso, o comando deve saber a posição de cada um de seus eixos. Algumas máquinas CNC ao iniciar o seu sistema, já possuem uma rotina de zeramento. No caso do robô utilizado, o Kuka KR5, esse não possui. Porém, como o mesmo possui uma bateria interna no comando para armazenar a posição de cada eixo, essa etapa não se faz necessária regularmente. Basicamente, essa etapa só se faz necessária quando o comando do robô perde a referência de um ou mais de seus eixos. Esse processo de referenciamento se denomina masterização dos eixos.

Um robô precisa ser masterizado nos seguintes casos:

- Após os reparos como a substituição de um motor.
- Se o comando do robô emitir uma mensagem indicando que a masterização é necessária. Isso se aplica a casos por exemplo em que as baterias internas estão vazias.
- Depois de trocar uma unidade de engrenagens.
- Depois de uma colisão, um impacto onde a parada final seja com uma desaceleração maior do que 250mm/s.

Zeramento da ferramenta: Ferramenta pode ser compreendida como o dispositivo a ser utilizado pelo robô. No caso desse trabalho, tanto o cabeçote fresador com sua fresa quanto o escâner são entendidos como ferramenta. Tanto o escâner quanto o cabeçote fresador devem possuir um TCP, que é o ponto da localização do centro da ferramenta perante as coordenadas da flange do robô e esse deve possuir uma orientação em relação a orientação das coordenadas da flange do robô. Essa etapa envolve os procedimento de zeramento e definição das orientações do TCP tanto do cabeçote fresador quando do escâner.

Zeramento da base: Nessa etapa, a base, ou dispositivo de fixação da peça a ser trabalhada deve possuir um sistema de coordenadas em relação ao robô. Essa etapa envolve o procedimento de zeramento desse sistema de coordenadas.

- **Ciclo do processo de escaneamento:**

Consiste no processo de escaneamento da peça para a obtenção do arquivo STL da peça a ser trabalhada, conforme ilustrado na Figura 4.3.



Figura 4.3: Ciclo do processo de escaneamento

**Fixação da peça:** Nessa etapa, a peça é fixada na mesa rotativa através de um dispositivo de fixação.

**Escaneamento:** Nessa etapa, uma trajetória de escaneamento deverá ser programada no robô e executada toda vez que se desejar obter a nuvem de pontos da peça a ser trabalhada.

**Arquivos STL:** Nessa etapa, ocorre a transformação do arquivo resultante do processo de escaneamento, que é uma nuvem de pontos para um arquivo no formato STL.

- **Ciclo do processo de alinhamento:**

Consiste no processo de alinhamento do modelo CAD com o modelo escaneado, conforme ilustrado na Figura 4.4.

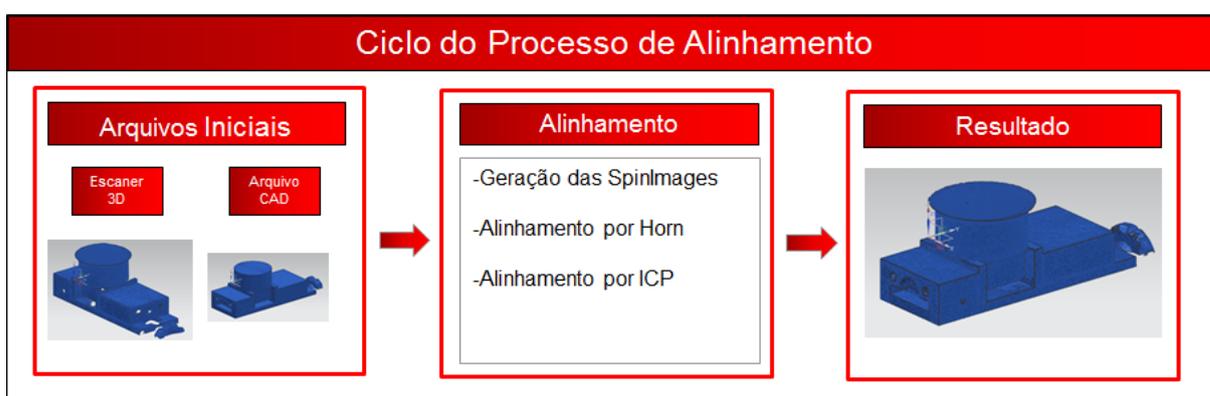


Figura 4.4: Ciclo do processo de alinhamento

**Arquivos Iniciais:** Como entrada dessa etapa, são os arquivos STL proveniente do processo de escaneamento, e o arquivo CAD do projeto da peça salvo no formato STL. A diferença dessa etapa para a anterior, é que a anterior ocorre num sistema

próprio de manejo do escâner. Esse, ocorre em um sistema computacional programado para esse trabalho.

**Alinhamento:** Nessa etapa, ocorre primeiro a geração das SpinImages de todos os pontos do arquivo escaneado e do arquivo do projeto da peça. Essas imagens são comparadas e encontrados três pontos em cada arquivo que sejam similares. Então esses pontos devem ser inseridos no algoritmo de Horn, que realizará o alinhamento inicial e então, deverá ser realizado o refinamento do alinhamento através do algoritmo ICP. Nessa tese, o método de Johnson, ou seja, SpinImages e o algoritmo de Horn são programados em Java, para sua devida comprovação. O método de ICP, por ser já um método difundido e utilizado por diversos softwares para refinamento do alinhamento de malhas, foi aplicado em um sistema comercial apenas para sua comprovação.

**Resultado:** Nessa etapa, como resultado, o arquivo da peça escaneada deverá ter o mesmo sistema de coordenadas do arquivo da peça projetada.

- **Ciclo do processo de reconhecimento de rebarbas:**

Consiste no processo de comparação entre os arquivos STL do modelo CAD com o escaneado e a separação das geometrias que deverão ser removidas, conforme ilustrado na Figura 4.5.

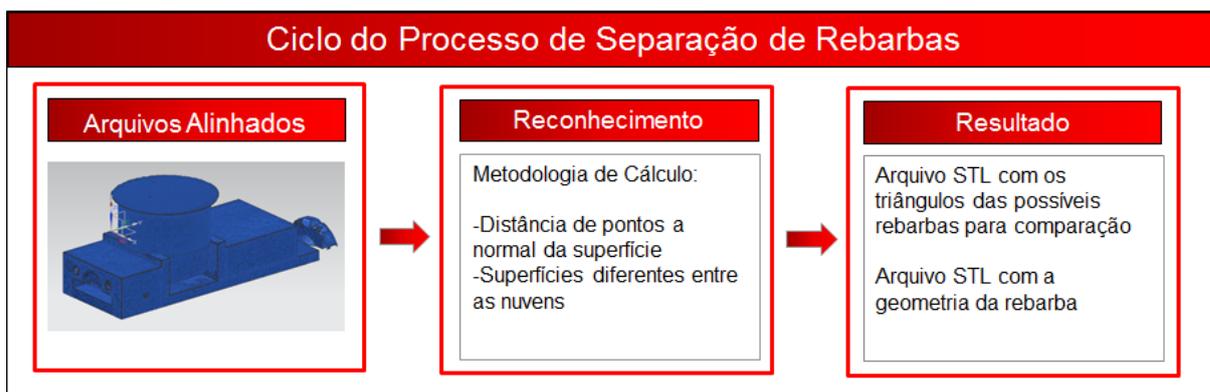


Figura 4.5: Ciclo do processo de separação de rebarbas

**Arquivos alinhados:** Como entrada dessa etapa, são os arquivos alinhados na etapa anterior. Ou seja, tanto o arquivo STL da peça projetada quanto o STL da peça escaneada com o mesmo sistema de coordenadas.

**Reconhecimento:** Nessa etapa, é aplicada a metodologia de reconhecimento proposta pelo trabalho de mestrado do autor dessa tese de doutorado 52.

Resultado: Como resultado, se obtém dois arquivos.

- Um arquivo no formato STL contendo as rebarbas, ou seja, do arquivo escaneado, todas as facetas ou pontos que não foram considerados rebarbas pela programação são apagados, ou apenas não salvos.
- Um arquivo no formato STL contendo as geometrias finais, ou seja, toda a rebarba separada do arquivo escaneado foi comparada a facetas do arquivo STL proveniente do CAD da peça projetada. Essas facetas do arquivo STL da peça projetada que tiverem rebarbas associadas no arquivo STL da peça escanada são separadas e salvas. As demais facetas são apagadas ou desconsideradas.
- **Ciclo do processo de geração de trajetórias:**

Consiste no processo de geração de trajetórias para a remoção das facetas consideradas rebarbas obtidas no ciclo anterior. A Figura 4.6 ilustra o ciclo de geração das trajetórias.



Figura 4.6: Ciclo do processo de geração de trajetórias

**Projeção de trajetórias:** Nessa etapa, trajetórias lineares deverão ser projetadas sobre as superfícies a serem removidas e o resultado matemático dessa projeção de retas sobre superfície serão as trajetórias. Trajetórias lineares devido a sua matemática ser mais simples que em trajetórias do tipo circular ou até *spline*.

**Pós-Processamento:** Essa etapa consiste em gerar os arquivos que o comando do robô reconhece como programa a partir das trajetórias geradas na etapa anterior. Essa fase também consiste em uma fase de verificação de colisões uma vez que é realizada em um sistema comercial.

Inserção do Código G: Nessa etapa, as trajetórias pós-processadas deverão ser transferidas para o comando do robô para serem executadas.

- **Usinagem Final:**

Nessa etapa, o robô executa a usinagem das geometrias caracterizadas como rebarbas, substituindo assim a atividade manual de rebarbação e comprovando a metodologia de geração da trajetória para um robô de forma automática, diminuindo assim ou eliminando a necessidade também de um programador para o robô.

## 4.2 Corpo de prova

Para os ensaios e validação do método proposto por essa tese, foram definidos dois corpos de prova.

Os requisitos para a definição dos corpos de prova são:

- O corpo de prova deve possuir geometrias caracterizadas como rebarba.
- O trabalho usou ferramentas disponíveis nos laboratórios das universidades Unimep – Santa Bárbara D'Oeste e USP – São Carlos para a realização do trabalho. Portanto, o corpo de prova deve possuir uma geometria que facilite a fixação do mesmo no dispositivo de fixação.
- Os corpos de prova devem ser testados em duas situações, uma geometria simétrica e uma geometria assimétrica para ensaiar os problemas e soluções dos métodos de alinhamento.

### 4.2.1 Corpo de prova A

O corpo de prova A consiste em uma geometria simples e simétrica. Pode ser comparada a rebarbação de uma roda automotiva. Apesar de que a geometria do corpo de prova A é mais simples, ele se torna mais complexo para o alinhamento quanto a questão da simetria e simplicidade no formato. Devido sua forma, o corpo de prova possui poucas variações em seu formato, isso acarreta em um número maior de imagens similares por um dos processos de alinhamento, o *SpinImages*. Esse maior número de imagens similares reforça a necessidade de um procedimento que consiga compará-las e definir corretamente as similaridades, contribuindo assim para a validação das soluções propostas. A Figura 4.7 ilustra o projeto do corpo de prova A.

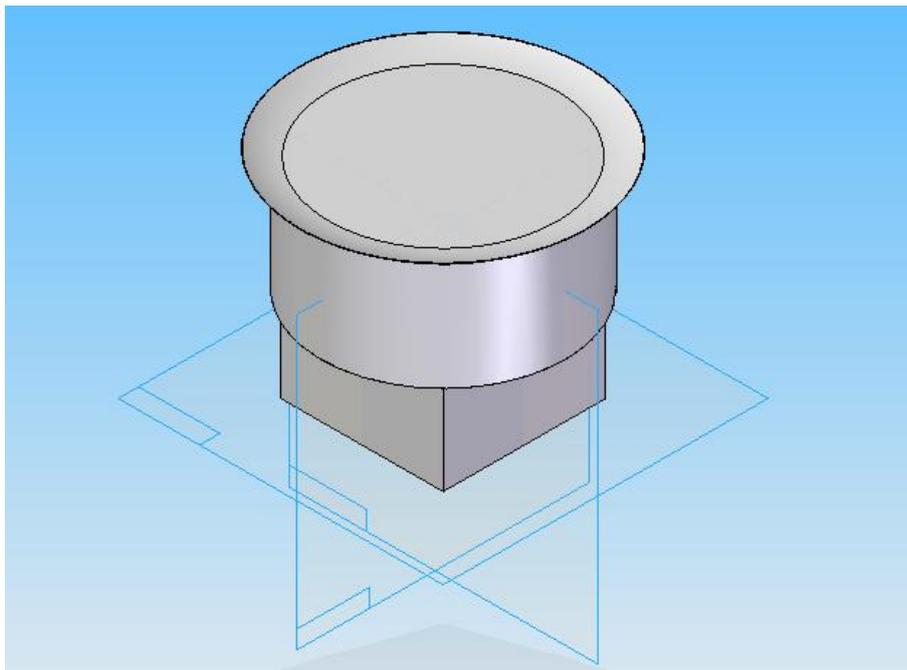


Figura 4.7: Modelo CAD do corpo de prova A

#### 4.2.2 Corpo de Prova B

O corpo de prova B consiste em uma geometria simples, assim como o A, também sem muitas variações em sua geometria como furos, ressaltos entre outros, porém, assimétrica, comprovando as diferentes imagens geradas pela metodologia *SpinImages*, assim como o processo de alinhamento como um todo. A Figura 4.8 ilustra o modelo CAD do corpo de prova B.

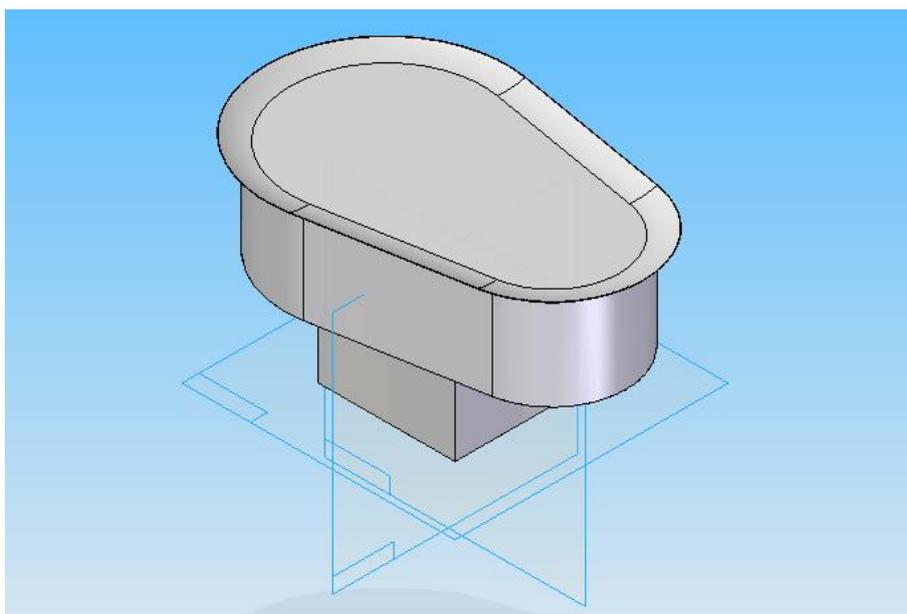


Figura 4.8: Modelo CAD do corpo de prova B

### 4.2.3 Projeto e Usinagem dos Corpos de prova

As fabricações dos corpos de prova consistiram em 6 principais fases. A Figura 4.9 e a Figura 4.10 ilustram essas fases.

Projeto CAD: O projeto CAD foi realizado no sistema CAD Solid Edge V20.

Trajectoria CAM: A trajetória da ferramenta foi realizada no sistema CAM NX9.

Dispositivo de fixação: Como o corpo de prova foi usinado em um centro de usinagem 3 eixos, houve a necessidade de duas usinagens em cada corpo de prova, uma de cada lado dos corpos de prova. Com isso há a necessidade de um novo zeramento após a inversão do lado do corpo de prova no centro de usinagem. Para garantir a maior precisão possível nos zeramentos dos dois lados, foi decidido por esquadrear o blank no próprio centro de usinagem, ou seja, para garantir que o blank possua o menor erro dimensional e de forma possível. Um bloco quadrado foi parafusado embaixo do blank permitindo assim que uma ferramenta realizasse o contorno da peça por completo.

Esquadreamento: Para a realização do esquadreamento, ou seja, para acertar o perpendicularismo das faces do blank, foram programadas manualmente direto no comando da máquina operações de faceamento no lado superior e inferior do blank e então, realizada uma operação de contorno.

Usinagem lado 1: Tendo o blank esquadreado, realizou-se o zeramento no centro de usinagem e iniciou-se a usinagem do lado inferior primeiro. Esse apresenta um bloco na parte inferior similar ao bloco parafusado para o esquadreamento. A diferença é que esse está em uma localização usinada em relação ao corpo de prova garantindo assim menores erros ao ser fixado dispositivo de fixação acoplado ao robô posteriormente.

Usinagem lado 2: Após a usinagem do lado inferior, a peça foi virada 180°, executando-se novamente o processo de zeramento antes de iniciar a usinagem da segunda face do corpo de prova.

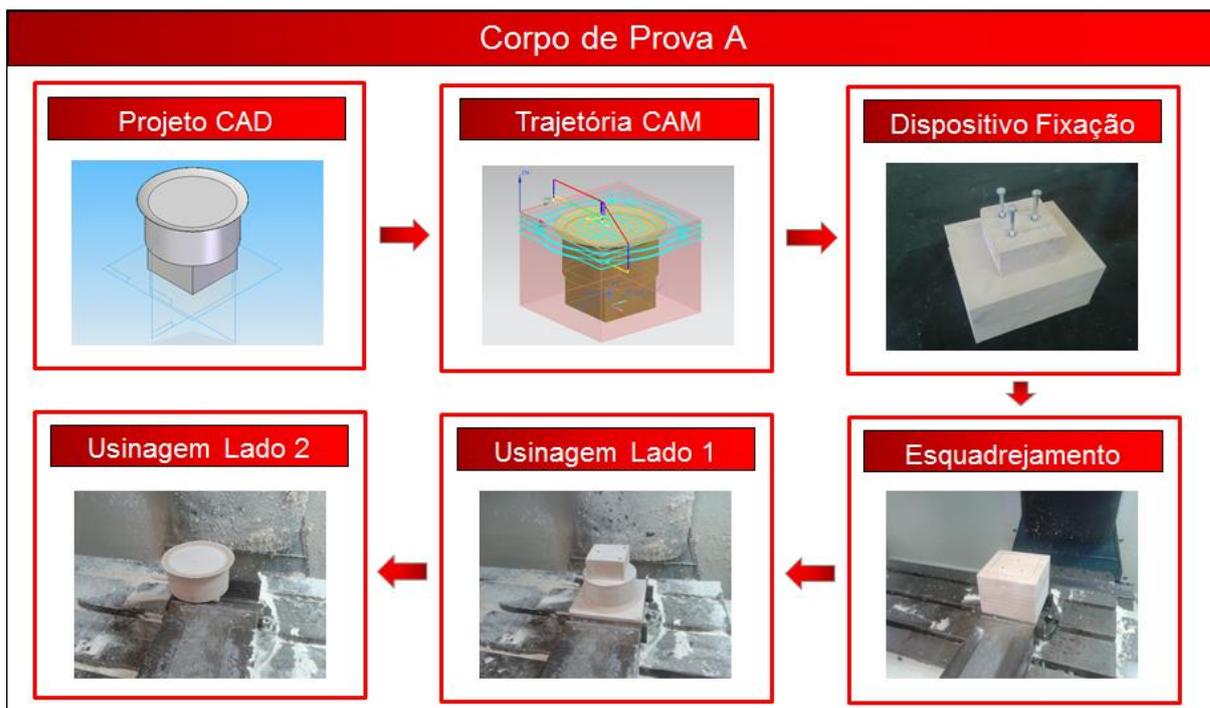


Figura 4.9: Projeto e manufatura do corpo de prova A

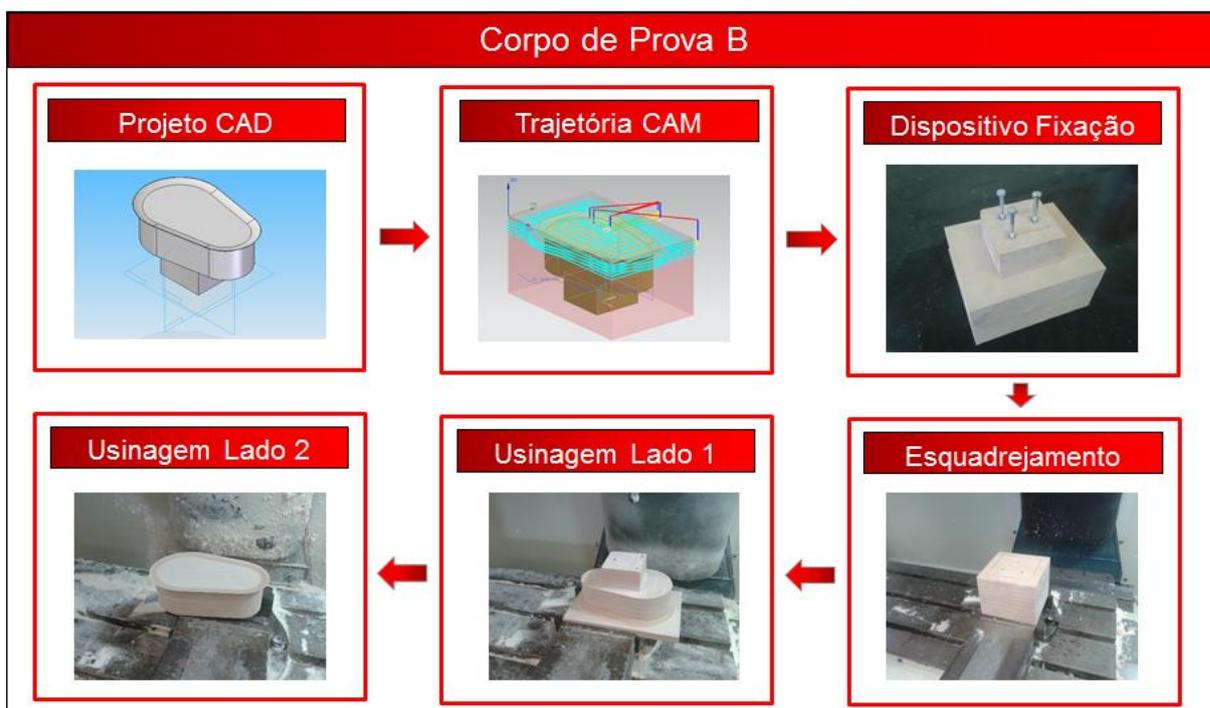


Figura 4.10: Projeto e manufatura do corpo de prova B

## 5 Implementação e Validação

Nesse capítulo é apresentado a implementação do método proposto. Como validação, a cada etapa implementada é apresentado os resultados da mesma e ainda, por último é apresentado os corpos de prova rebarbados pelo robô industrial.

### 5.1 Preparação do sistema

Esse item apresenta os procedimentos para preparar o sistema para a realização do trabalho. A sequência apresentada nesse capítulo é a mesma da apresentação geral do conceito apresentada no capítulo 4.

Antes de apresentar os dispositivos de fixação do escaner e de fixação do cabeçote fresador, é importante ressaltar que o método dessa tese sugere uma troca de ferramenta, ou seja, do escaner para o cabeçote fresador e vice-versa automática. O motivo dessa troca é que as duas ferramentas na flange do robô são inviáveis, por causa de dois principais motivos:

- Em relação ao cabeçote fresador, o fato do robô estar levando duas ferramentas, onde enquanto uma está em uso a outra está apenas junto significa uma massa desnecessária na flange do robô. Isso acarreta na redução da capacidade de esforços que o robô poderá suportar pelo cabeçote fresador durante a operação de rebarbação. Mais massa no braço robótico significa mais inércia durante os movimentos além de levar o robô perto do seu limite de carga. A capacidade do robô é fixa, portanto, quanto menos massa tiver em sua flange, mais esforços de trabalho o robô poderá realizar.
- Em relação ao escaner. Esse é um equipamento de grande valor e muito sensível a vibrações e choques mecânicos.

Porém, essa troca automática se torna viável em aplicações industriais. Como para essa tese, a sua principal função é proteger os equipamentos e justificar sua necessidade, foram desenvolvidos dois dispositivos separados de fixação, um para o escaner e outro para o cabeçote fresador. A sua troca nessa tese foi feita de forma manual.

#### 5.1.1 Fixação do Escâner

O Escaner utilizado no trabalho é uma câmera modelo *Hand Scan – ExaScan* da *Creafom*, uma empresa Canadense. Apresenta uma resolução de 0,05mm, uma

precisão de 0,04mm e faz sua leitura a uma distância de 300mm  $\pm$  150mm da peça a ser escaneada. Esse equipamento é ilustrado na Figura 5.1.



Figura 5.1: Hand Scan - ExaScan da Creaform

O software utilizado para gerar as nuvens de pontos através da câmera é o *VX Elements – 3D Digitalizing Software 1.0*.



Figura 5.2: Go Scan. Equipamento usado para escanar a ExaScan

Como o escaner é um dispositivo para uso manual, nesse não há furos ou flange para sua fixação no robô. Portanto, há a necessidade de projetar e construir um dispositivo de fixação que se adapte a empunhadura do escâner. Para isso, ocorre a necessidade de se ter a geometria da empunhadura do escâner assim como do

escâner por completo. Para o projeto do dispositivo de fixação, utilizou-se um segundo escaner, apenas para digitalização do escâner utilizado, uma câmera *GoScan*, Figura 5.2.

Através do segundo equipamento, a *GoScan*, gerou-se uma nuvem de pontos e posteriormente um arquivo STL do Escaner *Hand Scan – ExaScan* utilizado. Para facilitar o uso desse arquivo para o projeto e fabricação do dispositivo, o mesmo foi convertido para sólido no formato *Parasolid* através do software *GeoMágic*. A Figura 5.3 ilustra o arquivo e uma das telas do *GeoMágic*. Essa tela representa o arquivo STL do escaner já transformado em modelo sólido.

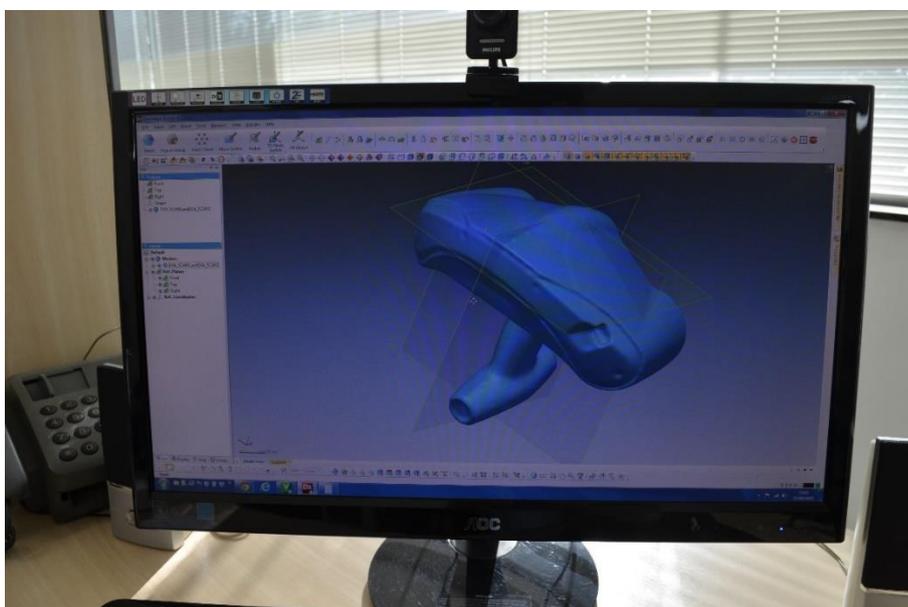


Figura 5.3: Arquivo STL sendo processado no Software *GeoMágic*

A justificativa para o uso de um segundo escaner diferente do primeiro para essa aquisição foi que tanto os escaners quanto os softwares foram fornecidos pela empresa *RGB Sistemas* em Santa Bárbara D'Oeste, SP. E essa empresa possui equipamentos diferentes para atender diferentes tipos de necessidades.

Para o projeto do dispositivo de fixação, requisitos considerados foram:

- Uso do menor volume de material possível. Para isso, o dispositivo buscou acompanhar a geometria do escâner.
- O dispositivo deve ser o mais leve possível. Isso porque o robô possui uma capacidade limitada de carga. O peso do escâner mais o dispositivo de fixação não pode ultrapassar 5 Kg. Portanto, o material definido foi o alumínio.

- O escâner é do tipo *Hand Scan*, ou seja, ele possui uma empunhadura para ser segurado e guiado manualmente pela peça. Uma vez que o escâner foi projetado para ser segurado por essa empunhadura, o dispositivo de fixação deve levar isso em consideração.
- Logo abaixo da empunhadura, o escâner possui o emissor laser, que faz a referência para as câmeras. Esse tem que ser deixado exposto.
- O escâner possui um botão de acionamento. Esse também foi deixado exposto para um possível projeto de dispositivo de acionamento automático ou manual mesmo como pode ser visto na Figura 5.4.

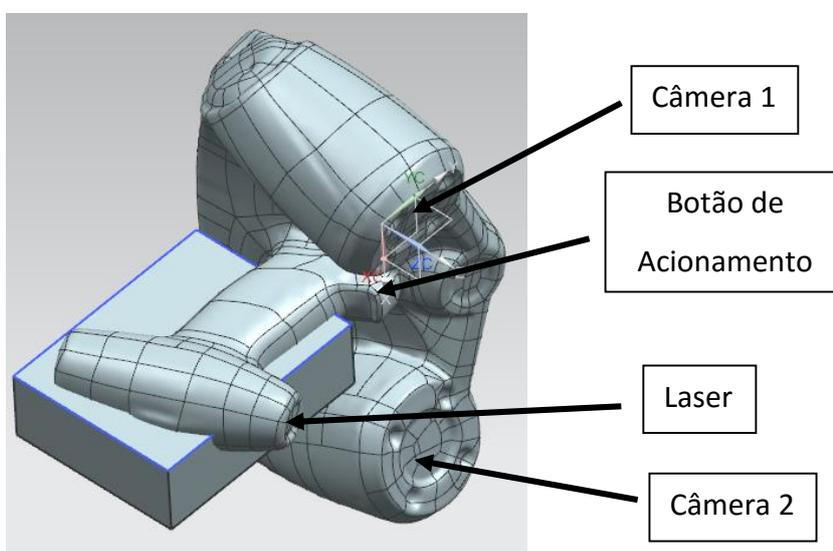


Figura 5.4: Detalhes do modelamento do dispositivo.

No software NX9, através do modelo sólido do escaner, foi definido um bloco de alumínio cujo tamanho compreendeu toda a empunhadura do escaner deixando para fora do mesmo o botão de acionamento e o emissor do laser. No modelo sólido do escaner foi realizado um comando que cresceu suas paredes em 3mm e em seguida foi realizado um comando de subtração da geometria do escaner aumentada de 3mm com esse bloco de alumínio, restando a geometria da empunhadura do escaner com uma folga de 3mm. Essa folga foi realizada para a colagem de uma fita de borracha para evitar riscos a empunhadura do escaner. Então, ainda no software NX9, foram realizadas as gerações da trajetória da ferramenta para esse ser usinado no centro de usinagem 3 eixos da Unimep.



Figura 5.5: Imagens sequenciais do desenvolvimento e fabricação do dispositivo de fixação do escaner

Após isso, com o dispositivo pronto, apenas se tem que montar o mesmo à flange do robô. A Figura 5.5 ilustra todo esse processo.

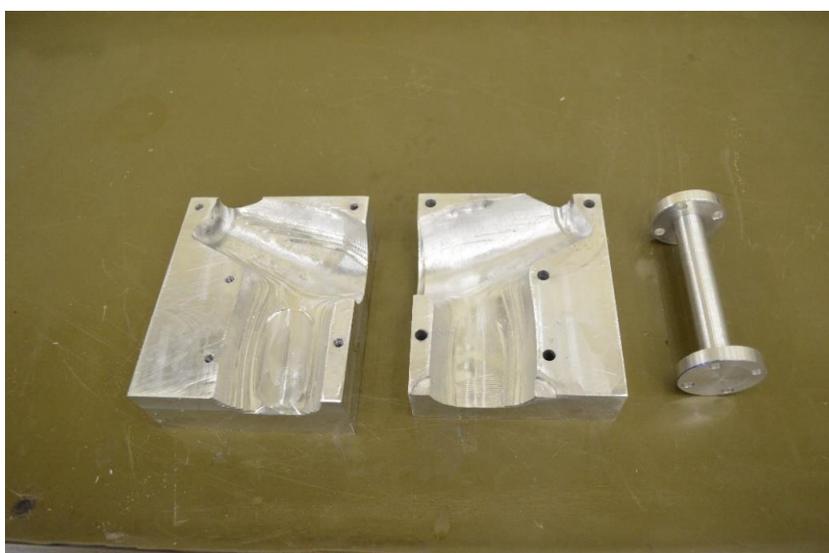


Figura 5.6: As duas partes que abraçam o escaner como dispositivo de fixação e uma flange de montagem e distanciamento do robô.

A Figura 5.6 ilustra as duas partes usinadas e uma terceira peça, uma flange; Essa flange tem duas funções, uma que é fixar o dispositivo de fixação do escâner a flange do robô, e também, distanciar o escâner do robô permitindo assim uma

maior amplitudes dos movimentos fazendo com que o escâner só colida com o robô em posições mais extremas. A fabricação dessa flange foi utilizando um torno convencional disponível no laboratório de processos de fabricação da Unimep

### 5.1.2 Fixação do Cabeçote Fresador

O robô e o *Spindle* foram gentilmente fornecidos pelo Professor Dr. Reginaldo Teixeira Coelho, do laboratório Numa, da USP em São Carlos. Esse laboratório já realiza alguns testes com o *Spindle* no robô e portanto, o mesmo já possuía um dispositivo de fixação. Após uma avaliação, o mesmo se mostrou eficiente, sendo assim, não mais necessário o projeto e fabricação, apenas a montagem do sistema. A Figura 5.7 ilustra a montagem.

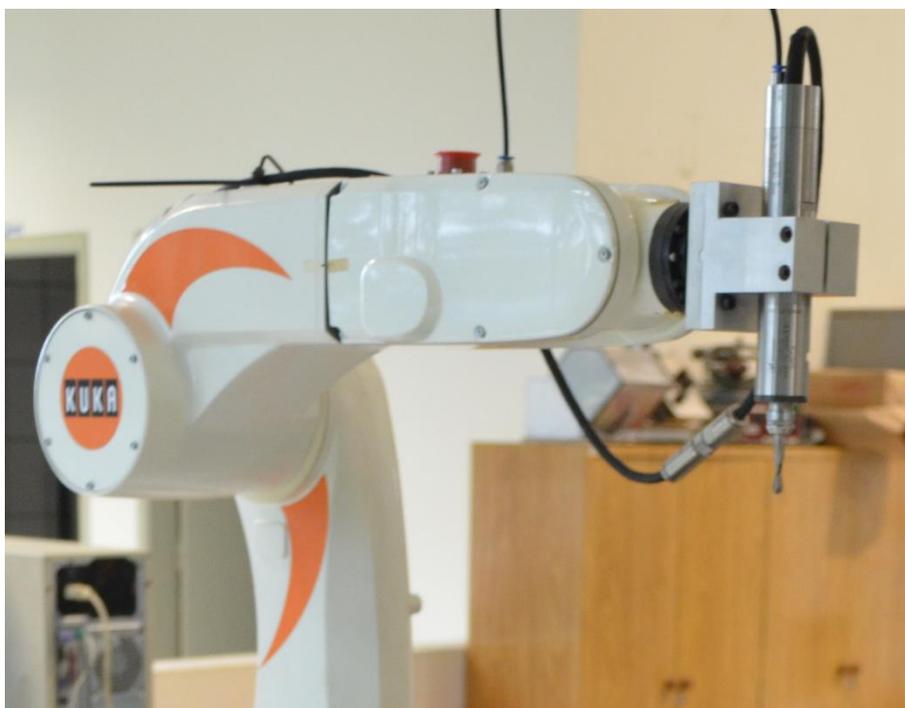


Figura 5.7: Imagem do Spindle montado no seu suporte e por sua vez, montado na flange do robô.

### 5.1.3 Mesa Rotativa

Conforme já discutido em artigos no capítulo 2, a mesa rotativa permite um maior acesso a peça tanto do processo de escaneamento como também para o processo de usinagem. Além do maior acesso, ela também permite que o robô trabalhe apenas de um lado do dispositivo de fixação, ou seja, não havendo a necessidade do braço ter que contornar a peça para alcançar o lado da peça oposto ao robô,

reduzindo assim a possibilidade de colisões do braço robótico com a peça e dispositivos de fixação.

Para a automação completa do processo, a mesa rotativa deverá ser automática, isso é, com um motor que permita a sua rotação. Ao programar o robô para fazer a trajetória de escaneamento da peça, a cada ciclo de escaneamento pode-se programar para o robô gerar um sinal e uma I/O do seu módulo de entradas e saídas comandando a mesa para girar, e também, aguardar um sinal da mesa dizendo que já está na posição correta para o próximo ciclo de escaneamento.

O laboratório Numa contém uma mesa rotativa suficiente para o projeto. Só que seu eixo de rotação é movimentado por uma alavanca manual, e não um motor. Como o objetivo dessa tese é apresentar um método com uma sequência de etapas e procedimentos que permitam e ajudem a automação do processo de rebarbação, foi decidido usar essa mesa manual, simulando durante os ensaios seu movimento na forma automática, permitindo assim a demonstração e comprovação do trabalho.

A Figura 5.8 ilustra a mesa rotativa fixada na mesa destinada para o robô, e ainda, em cima dessa mesa, uma morsa usada para fixação dos objetos a serem escaneados.

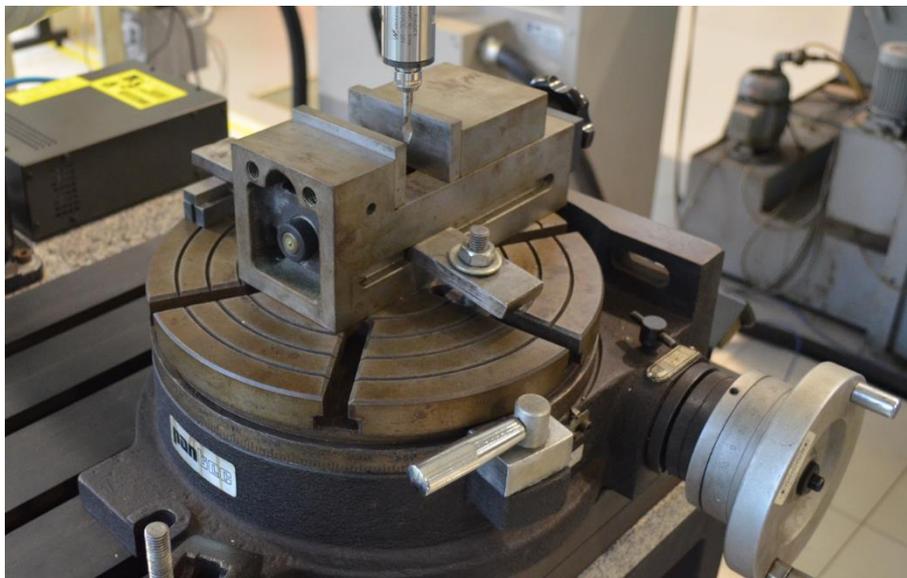


Figura 5.8: Fixação da mesa rotativa e da morsa.

#### 5.1.4 Zeramento do Robô

Essa etapa consiste em um procedimento para que o comando do robô saiba exatamente a posição de cada servomotor. Durante esse procedimento, também conhecido como Masterização, o robô é movimentado para posições de referência e os encoders salvam o valor da posição de cada eixo. Dessa forma, a posição de referência e o zero eletrônico são coincidentes.

Apenas um robô masterizado consegue ser movimentado e programado em coordenadas cartesianas.

Para a realização dessa etapa, o robô deve estar no modo de usuário Administrador. Apenas esse modo permite zerar o robô, ferramenta e base.

Essa etapa se fez necessária no Kuka KR5, pois devido ao fato de que o robô é utilizado para fins acadêmicos e podendo assim, ficar longos períodos de tempo sem utilização e assim, desconectado da energia, as baterias do comando apresentaram pouca carga e o comando do robô solicitou novamente a masterização dos eixos.

Após a recarga das baterias através do uso constante do robô, essa etapa não se faz mais necessária. Essa tarefa demandou tempo e energia para aprender o processo. Como a intenção da tese é contribuir com a comunidade científica, esse procedimento será descrito a seguir com imagens do posicionamento de cada eixo assim como os comandos necessários para a realização de tal tarefa.

O procedimento para tal realização está descrito no capítulo 9, em Anexo A.

#### 5.1.5 Zeramento da Ferramenta

Zerar a ferramenta, ou seja, encontrar o seu centro e sua orientação perante a flange do robô tem várias funções e necessidades. Dentre elas:

- Poder mover o robô na direção em que a ferramenta está no momento, e não sempre em relação as coordenadas globais ou de base.
- Quando se quer usinar com o robô, a necessidade de zerar ferramenta se faz da mesma forma de um centro de usinagem convencional. Para indicar ao comando onde a ferramenta está em relação ao seu zero absoluto.
- No robô, além de dizer a posição da ferramenta, se faz também necessário dizer a orientação da ferramenta. Em um centro de usinagem o *spindle* está em uma posição conhecida. No robô, o dispositivo de fixação do *spindle* pode ser

construído conforme o projeto da funcionalidade do robô, ou seja, a ferramenta pode ser fixada na flange do robô com a orientação que sua necessidade a torne necessária.

Os manuais do robô trazem essa etapa como sendo calibração da ferramenta. Dessa forma, será tratado como calibração do TCP.

Para calibrar a posição da ferramenta em relação as coordenadas da flange do robô, são necessários dois passos:

- Calibrar o TCP, que é o centro da ferramenta que se está utilizando no robô.
- Definir a orientação do sistema de coordenadas da ferramenta em relação ao sistema de coordenadas da flange do robô.

### **Calibração do TCP**

O método utilizado nessa tese para a calibração do TCP é o método XYZ – 4 Pontos. Esse método consiste em definir um ponto específico da ferramenta, no caso dessa tese, a ponta da fresa como o seu centro. Em caso de outras ferramentas, como por exemplo, uma tocha MIG de soldagem, pode-se definir o TCP como a saída do arame de solda. Em uma pistola de pintura, o centro do laque de pintura entre outras aplicações. Para essa definição, o comando do robô necessita que a ferramenta seja posicionada em quatro diferentes posições em relação a um ponto fixo no volume de trabalho do robô. Esse ponto fixo pode ser em qualquer lugar no volume de trabalho, desde que o robô consiga posicionar a ferramenta nessas quatro diferentes posições. A ponta da ferramenta tem que estar o máximo possível no mesmo lugar. O que se torna difícil quando ao movimentar o robô, para reposicioná-lo, esse centro sai de posição. Esse erro de posicionamento pode ser minimizado quando as posições que se calibra o TCP da ferramenta são as mais diferentes possíveis, porém, sempre há um erro, pelo mínimo que seja de posicionamento. A Figura 5.9 ilustra as posições usadas.

Para definir essas posições no comando do robô, deve ser selecionado a opção **Configure** → **Tool** → **Measure** → **XYZ 4-Point**. Em seguida, deve-se escolher o número da ferramenta, configurar os dados da mesma e então, após realizar as posições ilustradas na Figura 5.9, deve-se inserir no comando as coordenadas após cada posição, ou seja, após cada posição, antes de mudar para a próxima a necessidade de aceitar a posição atual no comando do robô.

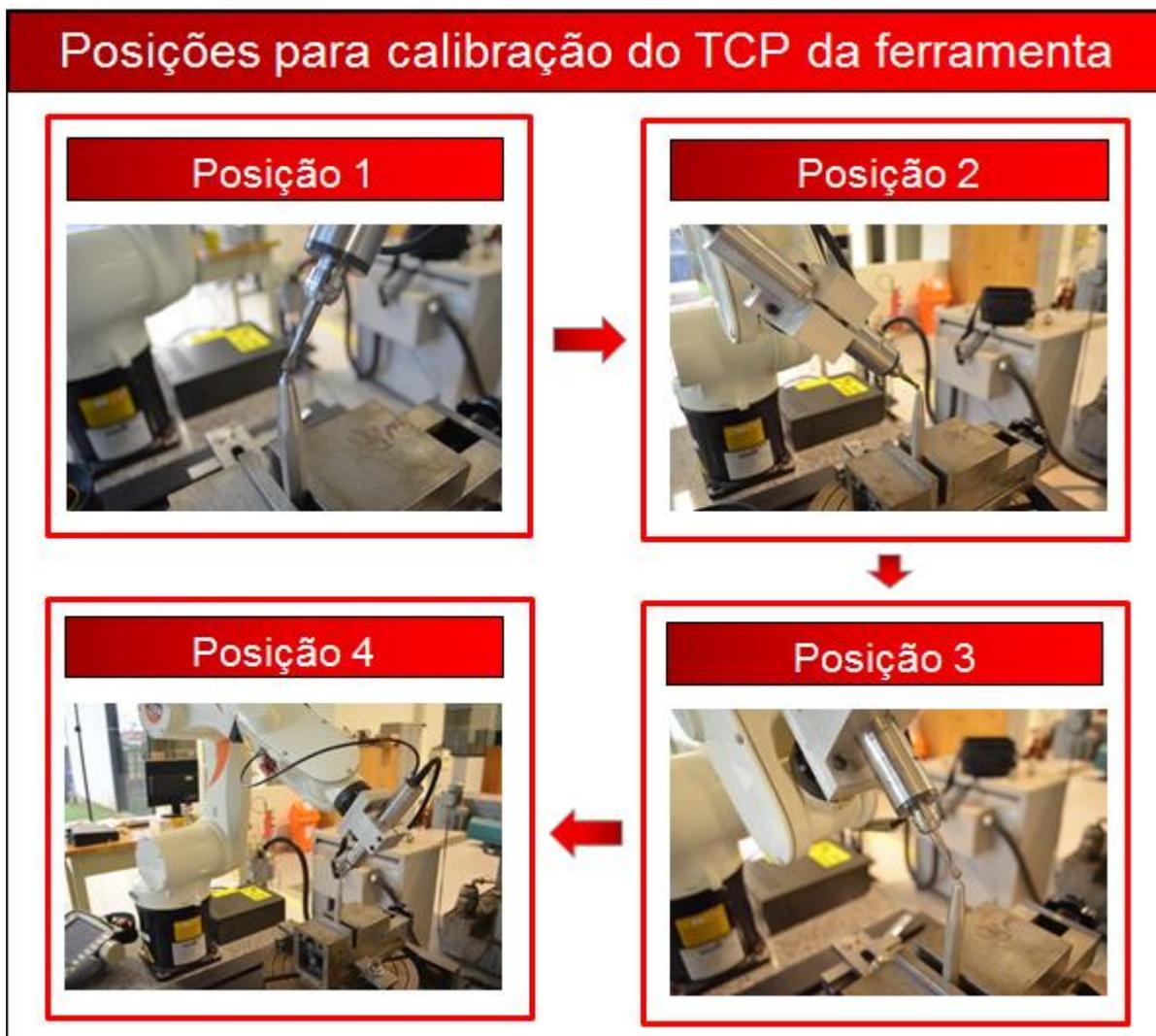


Figura 5.9: Sequencia de posições para a calibração do TCP da ferramenta

### Definição da Orientação do Sistema de Coordenadas da Ferramenta

O método utilizado nessa tese para a definição da orientação do sistema de coordenadas da ferramenta, foi o ABC – 2 – Pontos.

Para a execução desse método e garantir o alinhamento dos eixos da ferramenta com as coordenadas de base do robô, foi criado um novo arquivo de programa. Quando se faz isso, a primeira instrução que já vem pronta é a do PTP Home, que leva o robô para a posição e orientação inicial. Então, deve-se executar esse programa. Esse procedimento foi definido através de testes e experimentos no robô para encontrar a melhor forma de definir a orientação da ferramenta.

Com isso, o robô se encontra com a ferramenta em uma orientação conhecida. Seleciona-se o comando **Configure** → **Tool** → **Measure** → **ACB – 2 - Point**. Em seguida, selecionar no comando do robô a opção de movimentação sobre as

coordenadas globais do robô, e movido o mesmo na direção Z Global negativo. Isso representa o X positivo da ferramenta. Inserido no comando, deve-se então mover o robô na direção Y positivo sobre o sistema de coordenadas globais do robô, que representa o Y positivo da ferramenta. A Figura 5.10 representa a orientação do sistema de coordenadas da ferramenta (*Tool*) em relação a ao global (*World*).

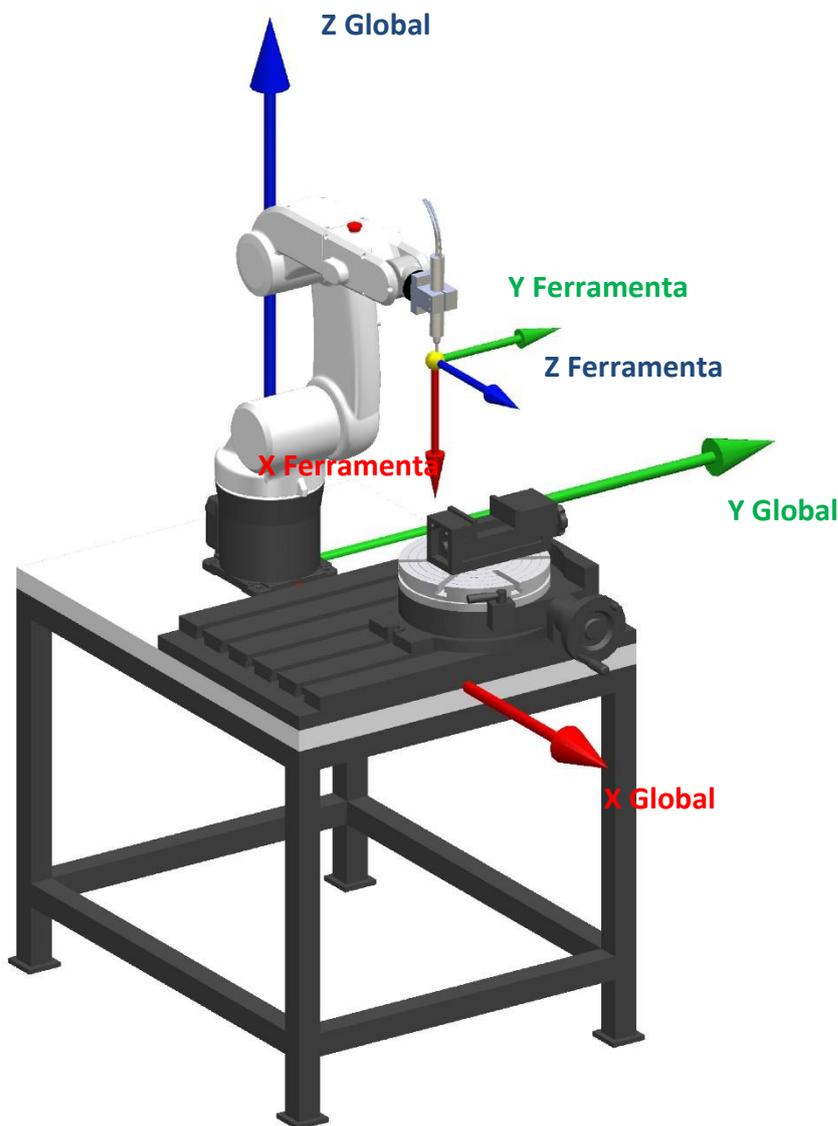


Figura 5.10: Sistema de coordenadas da ferramenta em relação ao global

A questão de precisão dessa orientação está diretamente ligada a precisão da fixação do cabeçote fresador no robô. Quanto mais precisos forem os posicionamentos dos furos do dispositivo de fixação mais preciso será sua orientação em relação ao sistema de coordenadas global do robô.

O escâner também precisa ser zerado como ferramenta. O encontro do laser com as duas câmeras poderia ser definido como seu TCP, ou seja, a região de leitura do escaner, que segundo o catalogo do equipamento está na faixa de 300mm do mesmo pode ser definido como o seu TCP e sua orientação, poderia ser arbitrária. No entanto, como o escâner trabalha a uma certa distância da peça e sua trajetória possui uma tolerância devido a ser um escâner manual, ou seja, o escaner pode realizar medições a uma distância de  $300\text{mm} \pm 150\text{mm}$  da peça, o TCP e orientação foi utilizado o do centro da flange. Esse atendeu as necessidades para a programação da trajetória do escâner.

### 5.1.6 Zeramento da Base.

Zerar base, assim como o zeramento de ferramenta, também tem várias funções e necessidades. Dentre elas:

- Poder mover o robô na direção em que a base está no momento, e não sempre em relação as coordenadas globais ou de ferramenta.
- Quando se quer usinar com o robô, a necessidade de zerar base se faz da mesma forma de um centro de usinagem convencional, ou seja, indicar ao comando onde a base, ou peça, ou dispositivo de fixação está em relação ao seu zero absoluto e portanto, onde a peça está em relação ao zero absoluto. No robô, além de dizer a posição da base, se faz também necessário dizer a orientação da base. Da mesma forma que a ferramenta. A base pode estar em qualquer posição no volume de trabalho do robô e ainda, em qualquer orientação.

O método utilizado nessa tese para a calibração da base é o Base 3-Point. Esse método consiste em definir na base 3 pontos. O primeiro ponto define a origem do sistema de coordenadas da base. O segundo, a direção do eixo X. Para definir o eixo Y, basta mais um ponto definindo o plano formado pelos eixos X e Y. Através desse ponto, define-se o plano e com o plano e a origem, define-se o eixo Y. Com os eixos X e Y se define o eixo Z.

O comando selecionado para a calibração da base é o **Configure** → **Base** → **Measure** → **3 – Point**.

A Figura 5.11 ilustra o local desejado para o zeramento da base, e os pontos necessários para que esse lugar seja reconhecido pelo comando do robô.

O ponto 1 definiu a origem do sistema. Ele é no vértice da morsa. O ponto 2 definiu a direção do eixo X. Ele foi definido em qualquer lugar da aresta da morsa apontada na figura. O ponto 3 definiu o plano XY, e portanto o eixo Y e o eixo Z. Ele foi definido em qualquer lugar no plano apontado na figura.

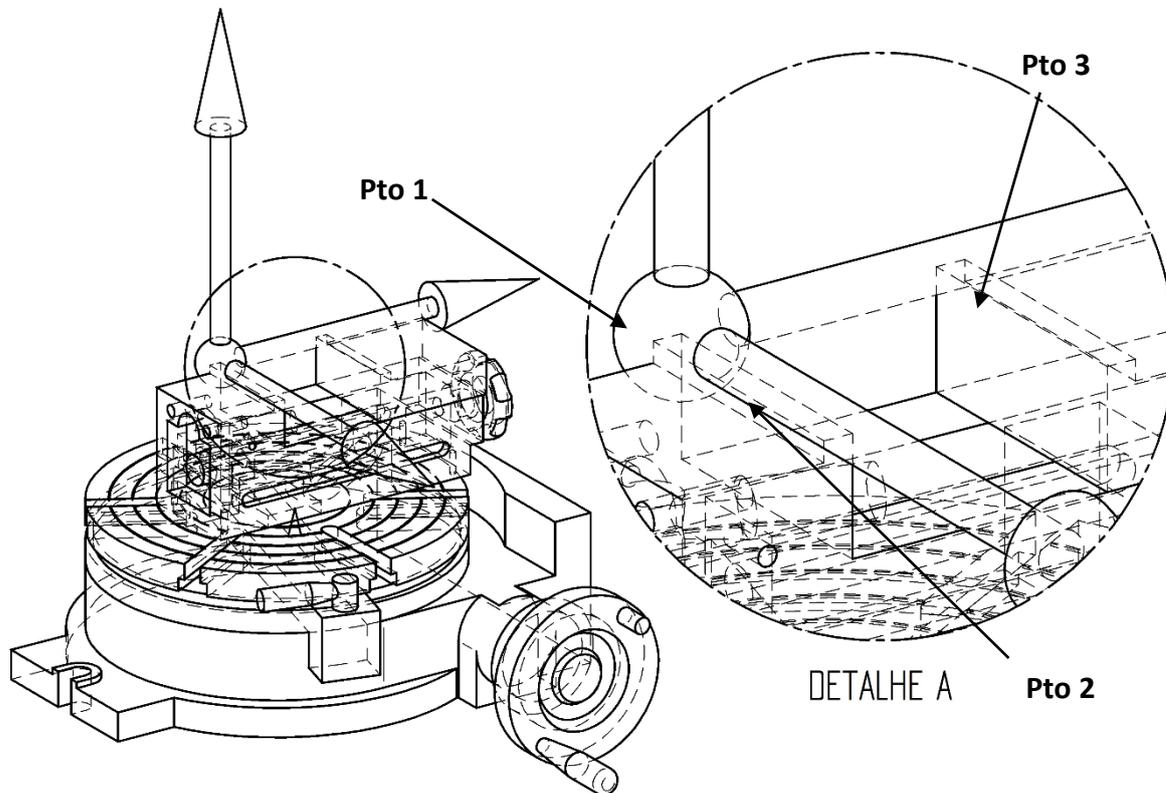


Figura 5.11: Zeramento da base

A Figura 5.12 ilustra as posições usadas no trabalho para a calibração da base com o uso do robô. São as mesmas da Figura 5.11, só que na planta.



Figura 5.12: Sequencia de posições para a calibração da base

A Figura 5.13 ilustra o robô com seus zeros definidos. A coordenada global, a da ferramenta e por último, a de base. É importante ressaltar que essas coordenadas

estão nessa posição pois para esse trabalho chegou-se à conclusão que esses foram os melhores lugares para zerar a ferramenta e principalmente a base.

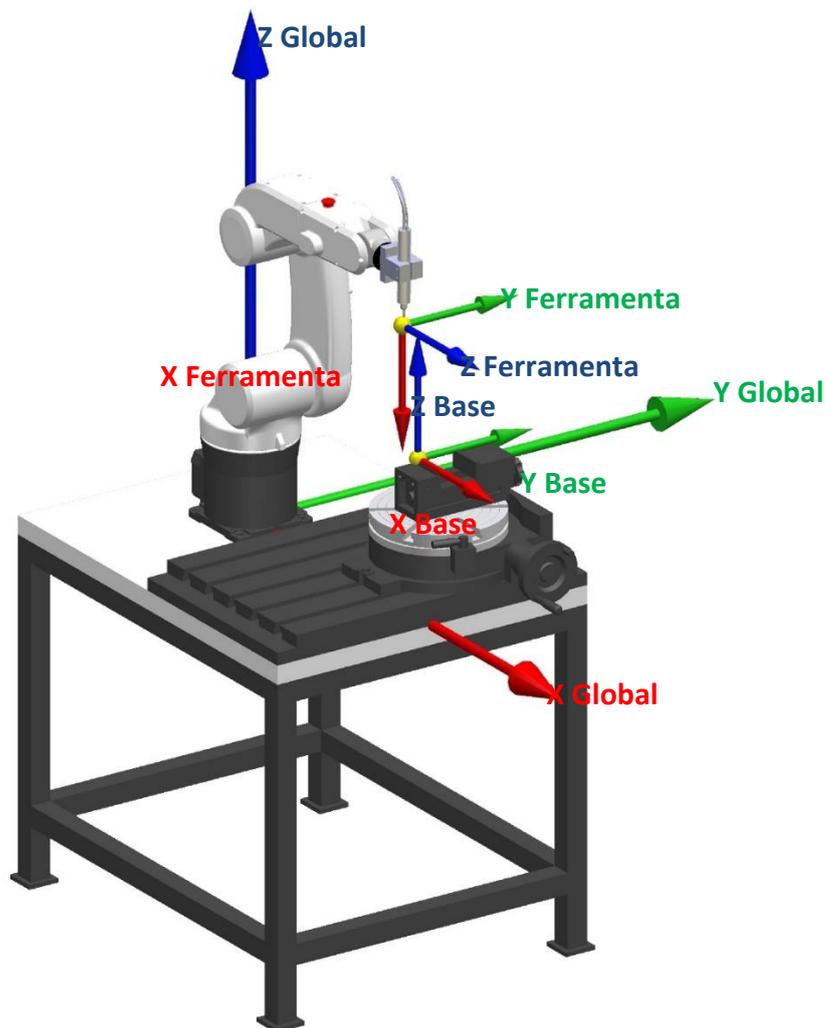


Figura 5.13: Sistema de coordenadas da base em relação ao da ferramenta e ao global

## 5.2 Processo de Escaneamento

Para a realização do processo de escaneamento, foram definidas 3 etapas. São elas fixação da peça, escaneamento e então, geração do arquivo STL da nuvem de pontos.

### 5.2.1 Fixação da peça

Como dispositivo de fixação da peça, foi usada uma morsa. E essa fixada sobre uma mesa rotativa.

O zeramento da peça pode ser feito ou na própria peça, ou no dispositivo de fixação. Contanto que o mesmo seja igual ao do processo de geração da trajetória

de usinagem, a sua escolha pode ser arbitrária. No caso desse trabalho, esse zero, que é o mesmo da peça modelada no CAD, é no vértice da morsa, conforme já apresentado.

Independentemente se o zero da peça for zerado na base, ou na própria peça, quando fizer a troca de peça para a realização da rebarbação da próxima, a precisão do posicionamento do processo está diretamente ligado a repetibilidade da fixação da peça na morsa. A morsa utilizada consegue garantir apenas dois posicionamentos através do processo de encostar a peça no dispositivo. Conforme ilustrado na Figura 5.14, o eixo Z é garantido quando a peça toca no mordente da morsa na direção Z e o eixo Y quando a peça toca na morsa na direção Y. O eixo X, não tem lugar para tocar. Poderia ser feito um dispositivo para o mesmo, mas nesse trabalho, foi utilizado de um paquímetro pré-ajustado para 15mm, onde o mesmo encostou na borda de fora da morsa, conforme indicado e em seguida tocou na peça.

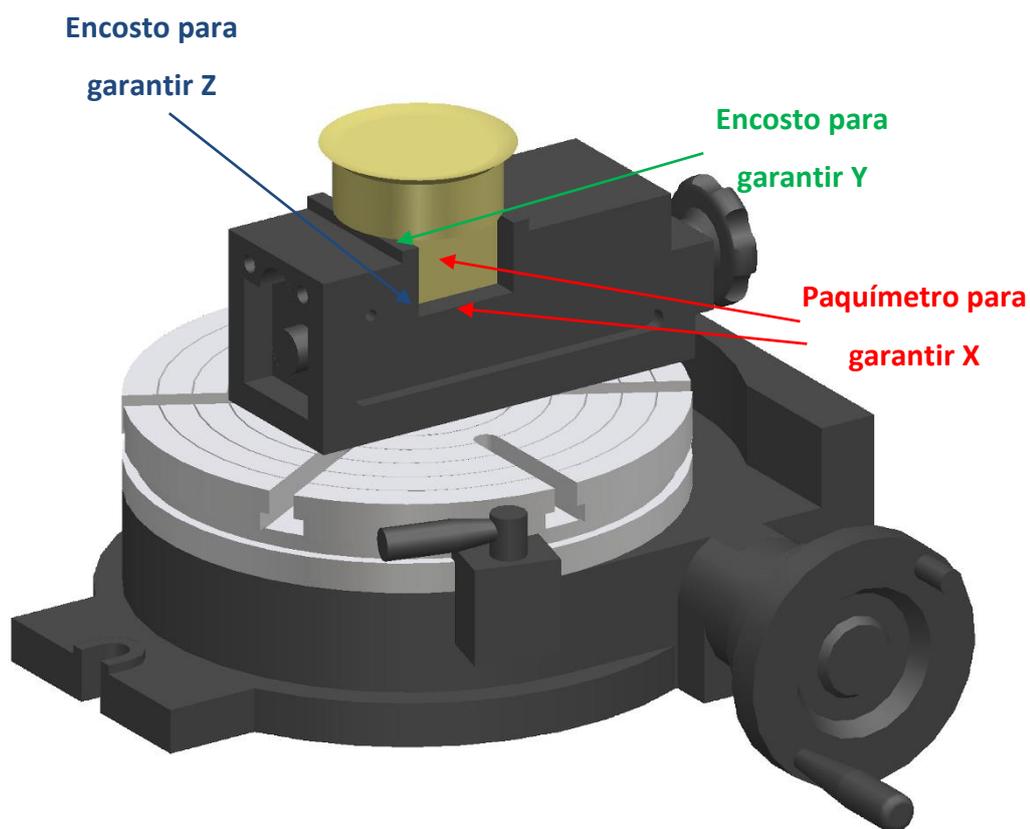


Figura 5.14: Fixação dos corpos de prova na morsa

A Figura 5.15 ilustra a fixação do corpo de prova A e a Figura 5.16 ilustra a fixação do corpo de prova B da forma que está ilustrado na Figura 5.14.



Figura 5.15: Peça A colocada no dispositivo de fixação



Figura 5.16: Peça B colocada no dispositivo de fixação

### 5.2.2 Escaneamento

A automação do processo de escaneamento consiste no robô guiar o escaner através da peça ao invés de um operador.

Para isso, uma trajetória precisa ser programada no robô para que o mesmo execute tal tarefa. Conforme descrito no Capítulo II, existem basicamente duas formas de se programar um robô: *On-line* e *off-line*.

A programação on-line, ou seja, através do *Teach Pendant*, as coordenadas da trajetória são salvas de acordo com o sistema global de coordenadas do robô, e para modificar as mesmas, tem que levar o robô para as posições que se deseja alterar uma a uma novamente. A vantagem está em que é uma programação mais rápida, pois necessita apenas do robô para tal execução. A desvantagem, é que se a mesa rotativa mudar de posição em relação ao volume de trabalho do robô, essa trajetória necessita ser reprogramada.

A programação *off-line*, ou seja, através de um sistema CAM por exemplo, as coordenadas são salvas em relação a um ponto comum na mesa rotativa, igual realizado para zerar ferramenta ou base. Então, deve-se usar um sistema para pós-processar como o CAMROB, onde o mesmo pode ainda simular a trajetória antes da inserção no robô. Para pós-processar, o CAMROB utiliza das coordenadas do zero ferramenta e do zero base, que devem ser fornecidos para o sistema previamente. Assim, se mudar a mesa e portanto a morsa de lugar, bastaria zerar a base novamente e toda a trajetória estaria modificada em relação a nova posição de base, não sendo necessário uma nova programação da mesma.

Nessa tese, como os testes foram feitos em uma mesma posição da mesa rotativa, a programação foi realizada de forma *on-line*. Devido ao fato de que a mesa é rotativa, foi realizado um programa no robô no qual o mesmo faz uma varredura em forma de arco em um único sentido da peça, conforme mostra a Figura 5.17, indicado pelo vetor na cor amarela. O robô parte de sua posição zero, e vai rapidamente para o início do arco amarelo, na velocidade máxima do equipamento. Do início ao fim do arco o mesmo realiza a trajetória em uma velocidade baixa, suficiente para o escâner adquirir a nuvem de pontos da área escaneada. Essa velocidade se deu através de testes. Foi-se aumentando a velocidade do robô até o ponto em que a aquisição da nuvem fosse prejudicada. Foi então utilizado 50% dessa velocidade máxima, ou seja, 1m/min. Do fim do arco, o robô volta ao início do arco por 6 vezes e então, após a sexta vez, o mesmo retorna para a posição zero. A cada vez que o robô sai do fim do arco para o seu início, a mesa gira 60°. Esse grau de giro também foi encontrado através de experimentos. Foi-se aumentando esse grau até o ponto em que a aquisição fosse comprometida. 60° foi o valor que os testes indicaram ideal. O número de repetições, que são de 6 vezes, se dá pela divisão de 360° por 60°. O arco verde representa o sentido de giro da mesa rotativa. As trajetórias de aproximação e retorno não estão indicadas na figura.

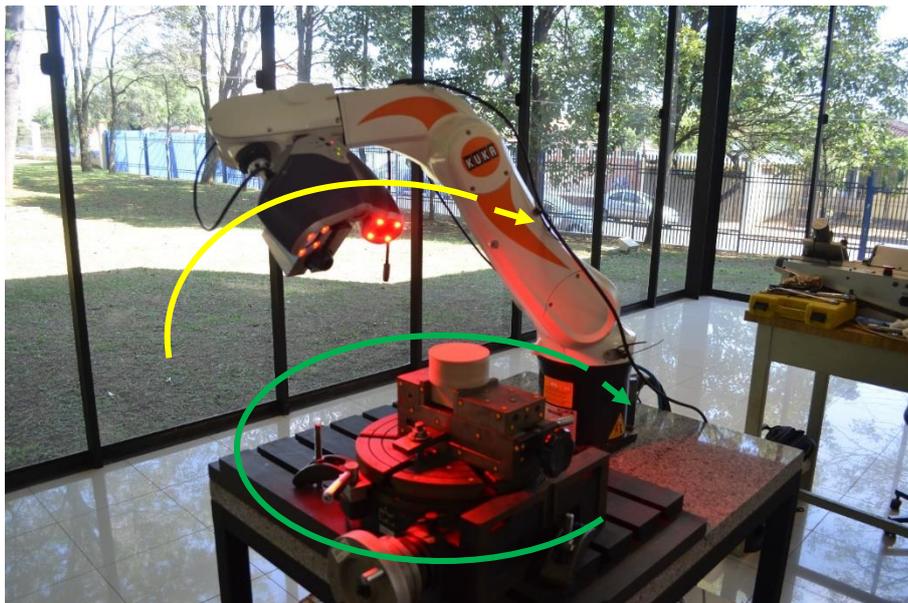


Figura 5.17: Robô guiando o escaner através do corpo de prova.

### 5.2.3 Geração do Arquivo STL

O software que controla a câmera é o *VX Elements – 3D Digitalizing*. Esse software, após o fim da digitalização gera automaticamente o resultado do escaneamento em um arquivo STL. Esse arquivo pode ser tratado no próprio software para fechamento de buracos ou redução da quantidade de pontos. Porém, como a câmera passou em todos os lugares de interesse nos dois corpos de prova, apenas foram salvos os arquivos com tamanho de aproximadamente 3Mb. Esse tamanho de arquivo se deu pelo tamanho das facetas. Com 3Mb as facetas geradas foram suficientes para representar as geometrias de rebarbas.

Como resultado do processo de escaneamento, tem-se um arquivo formato STL para cada um dos corpos de prova.

A Figura 5.18 ilustra o arquivo do corpo de prova A e a Figura 5.19 ilustra o arquivo do corpo de prova B. Os arquivos foram abertos no sistema CAD NX9 para permitir a visualização do resultado do processo de escaneamento. Os buracos vistos nas imagens estão em regiões de não interesse, e portanto, não são relevantes. É de responsabilidade do programador da trajetória do escaner conhecer do equipamento e da peça que será escaneada para realizar uma trajetória para o robô de forma a cobrir todas as regiões que contém as rebarbas. No caso dos corpos de prova, todas as geometrias dos mesmos estão escaneadas.

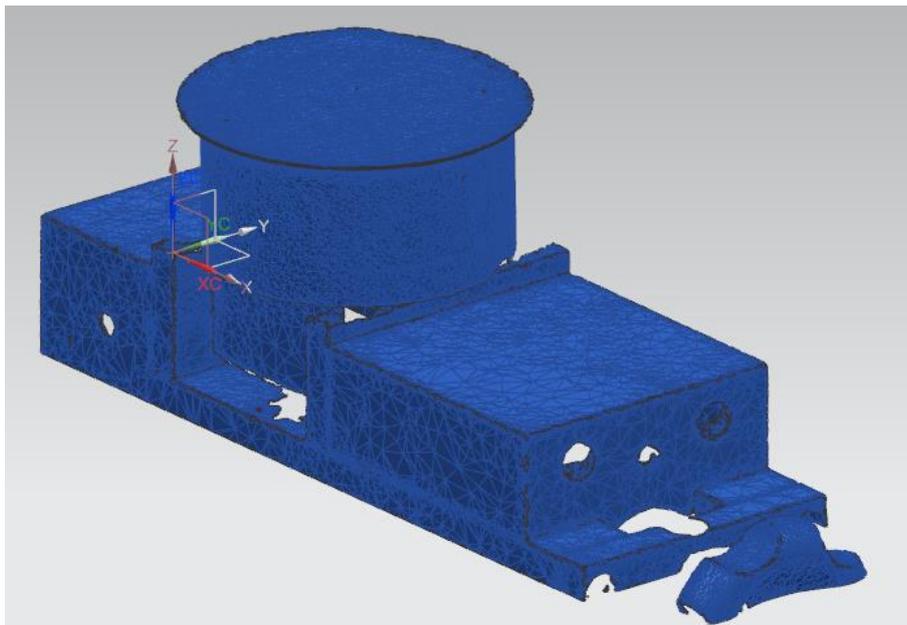


Figura 5.18: STL resultante do processo de escaneamento do corpo de prova A

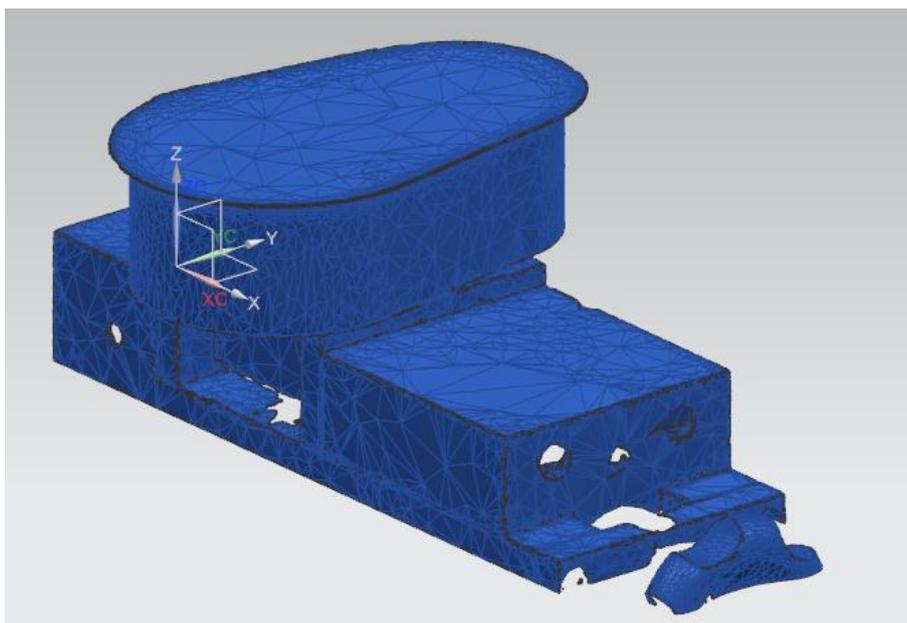


Figura 5.19: STL resultante do processo de escaneamento do corpo de prova B

### 5.3 Processo de Alinhamento

Uma vez escaneada a peça a ser rebarbada, essa precisa ser comparada ao seu modelo CAD. As regiões que contêm as rebarbas assim com suas geometrias e volumes tem que ser identificados para então definir as trajetórias da ferramenta do robô e ainda por fim, remove-las através do processo de usinagem denominado de rebarbação.

Um dos problemas em realizar essa operação constituída de uma série de métodos e etapas é o fato de que os sistemas de coordenadas do escâner ser diferente do sistema de coordenadas do modelo CAD, que por fim, ainda é diferente do sistema de coordenadas do robô. O que isso quer dizer: Um exemplo simples é solicitar para dois modeladores modelarem um paralelepípedo em um sistema CAD. Se for dito como parâmetro para os mesmos o comprimento, largura e altura, um deles pode modelar com um dos vértices coincidente no centro do sistema de coordenadas e ainda, definir o eixo X como sendo o comprimento do paralelepípedo. Já o segundo modelador pode realizar a mesma tarefa tomando o eixo Y como sendo o comprimento. Dessa forma, os dois modelaram sobre um mesmo sistema de coordenadas, porém, ao abrir os dois arquivos em um mesmo ambiente, eles não estarão sobrepostos. Isso porque sua orientação e translação não bate. Para isso, se faz necessário procedimentos matemáticos para trocar o sistema de coordenadas e um dos arquivos e fazer coincidir com o mesmo do outro arquivo.

Através de experimentos, identificou-se que o robô pode ter um zero conhecido e determinado pelo programador denominado de zero Base. O zero Base dessa tese foi definido em um dos dois vértices do mordente fixo da morsa usada como dispositivo de fixação. Conforme já demonstrado no trabalho.

Como a peça a ser rebarbada possui um modelo CAD. E essa vai ser comparada ao arquivo escaneado. O problema da diferença do zero do arquivo CAD com o do robô pode ser resolvido de forma simples, apenas modelando a peça de acordo com o zero igual ao definido no sistema de fixação do robô. Isso é, se define um ponto estratégico ou na peça ou no dispositivo de fixação que permita um processo de zeramento no robô. Então, ao modelar a peça no sistema CAD, essa deve ter seu zero no mesmo lugar e orientação que foi escolhida na planta, no robô e seus dispositivos. Dessa forma, ambos terão o sistema de coordenadas no mesmo lugar.

A Figura 5.20 e a Figura 5.21 ilustram o corpo de prova modelado no sistema CAD NX e salvo no formato STL.

O zero do processo de escaneamento é de certa forma não previsível, ou melhor, não se consegue determinar o mesmo em um lugar com precisão. O processo de criação do zero de referência do processo de escaneamento se baseia no procedimento que ao se pressionar o botão para escaner, o sistema identifica as etiquetas de referência na peça e cria o sistema de coordenadas, conforme

ilustrado na Figura 5.22. Ou seja, dependendo da região e orientação que o escaner for apontado para a peça, seu zero será em uma posição e orientação diferente. Porém a malha adquirida não depende do zero, mas sim da geometria da peça assim como a acessibilidade do escaner a peça.

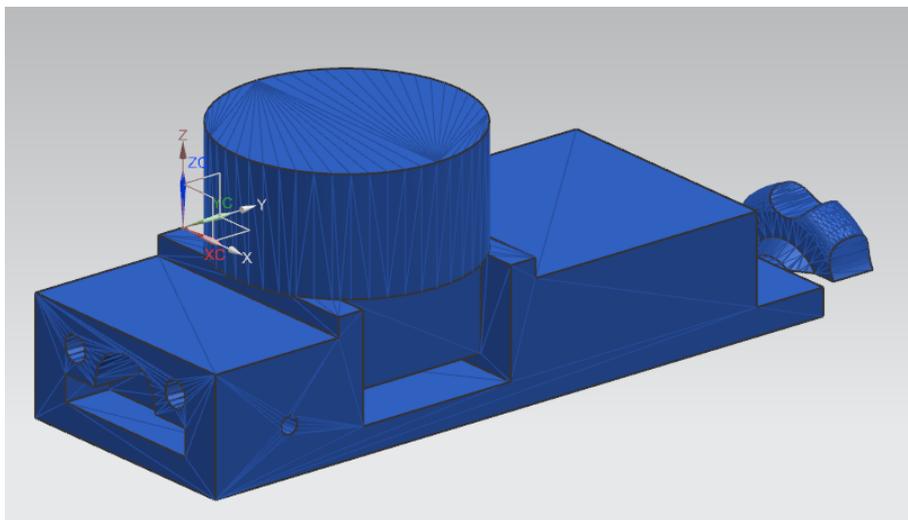


Figura 5.20: Corpo de prova A, modelado no sistema CAD NX e Salvo em formato STL

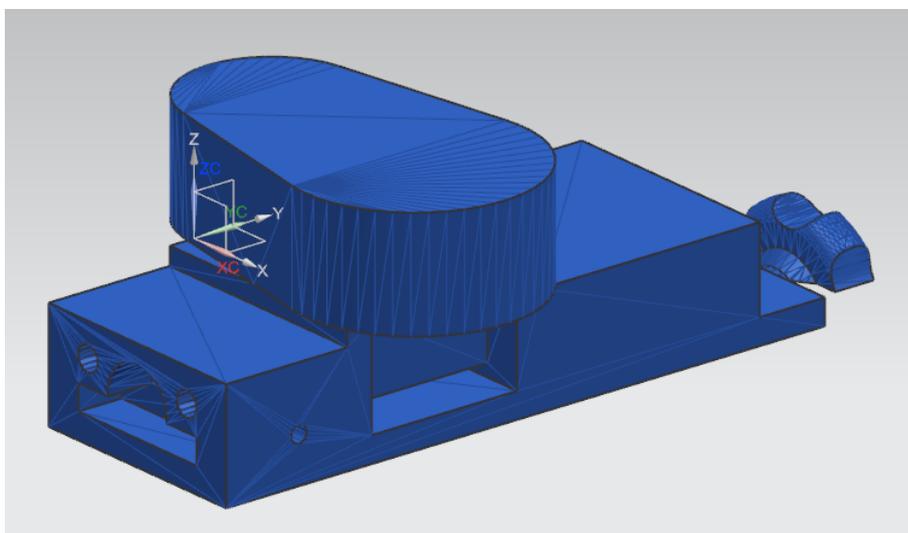


Figura 5.21: Corpo de prova B, modelado no sistema CAD NX e Salvo em formato STL

Na Figura 5.23, pode-se observar o sistema de coordenadas. Esse está em um lugar diferente do definido na peça projetada. Nessa figura, ainda se pode observar que o sistema tem um indicativo de distância do escaner da peça. Esse indica se o escaner está em uma distância considerada boa da peça, muito perto ou muito longe. Esse indicador ajuda no momento de programar a trajetória de escaneamento. E ainda, nessa mesma figura pode-se observar as etiquetas de

referenciamento. As brancas são as já reconhecidas pelo sistema, porém o escaner não as alcança no momento, assim como as em vermelho, além de já terem sido reconhecidas, o escaner está escaneando justamente na região onde elas se encontram.

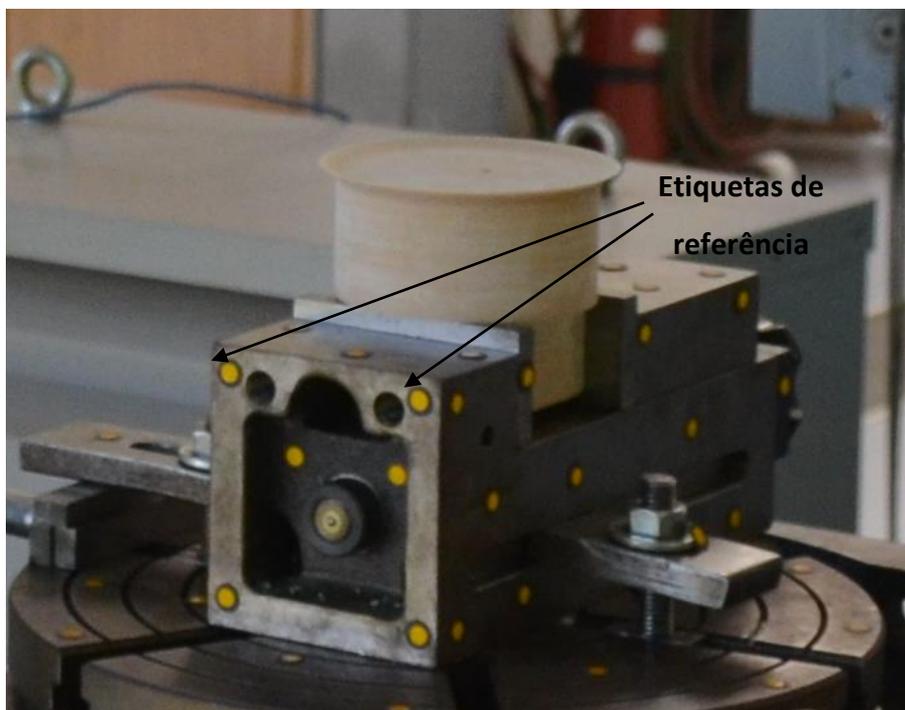


Figura 5.22: Etiquetas de referência do escaner

Dessa forma, a necessidade de métodos matemáticos para realizar o alinhamento é justificado, ou seja, se faz necessário uma sequência de processos de alinhamentos.

Conforme identificado na literatura, o melhor processo é o conhecido como ICP, que está descrito no capítulo 2, que interage todos os pontos da malha A com o da malha B. Porém esse processo tem uma restrição. As malhas têm que estar de certa forma pré-alinhadas, para que a matemática do processo possa interagir os pontos uns com os outros e alinhar as malhas.

Pesquisas na literatura também apontaram o método conhecido como algoritmo de Horn como um método eficaz para alinhamento de nuvens de pontos mesmo essas estando em posições totalmente arbitrárias, cujo algoritmo também está descrito no capítulo 2.



Figura 5.23: Comportamento do escaner e suas etiquetas

Em uma primeira leitura, se entende que o método de Horn é o mais indicado e melhor. Porém, umas de suas restrições para o perfeito alinhamento das malhas é que elas sejam simétricas ao número de pontos e formato, ou melhor, os pontos fornecidos para a sistemática têm que ser simétricos em ambas as malhas. Na Figura 2.47 pode-se observar essa restrição. Quando mesmo pontos são identificados em ambas as malhas, o processo matemático proposto por Horn proporciona um alinhamento perfeito. Essa restrição não pode ser alcançada uma vez que o processo de escaneamento possui tolerâncias e restrições. Um exemplo claro é que em um arquivo modelado, tem-se quinas que podem ser escolhidas como pontos para o algoritmo. Já em uma peça escaneada, o processo de escaneamento adquire pontos pertos da aresta, mas não a forma perfeitamente. Esse erro, pelo menor que seja já pode implicar em um erro significativo no alinhamento das malhas. Por isso, pode-se usar Horn primeiro para o pré-alinhamento seguido de ICP.

Ainda falando em restrições de Horn, outra restrição é que a interação do método depende do fornecimento de ao menos e não mais do que necessários 3 pontos em ambas as malhas. Porém, esses pontos têm que ser similares. Se foi escolhido um vértice na malha A esse mesmo vértice deve ser escolhido na malha B.

Para encontrar esses 3 pontos de forma automática, isso é, sem a necessidade de um operador de CAD, ou seja, uma pessoa que abra a malha escaneada assim como a malha que representa a peça modelada e manualmente selecione pontos e anote suas coordenadas e forneça para o algoritmo de Horn, foi encontrada na literatura o método de Johnson. Denominado de *Spin-Images*.

Portanto, para o alinhamento das malhas, esse trabalho aplica a sequência de métodos:

**Método de Johnson → Algoritmo de Horn → ICP**

### 5.3.1 Aplicação do Método de Johnson

Conforme descrito no capítulo 2, o método de Johnson se baseia em que para cada ponto da malha, por ser um arquivo STL, esse possui um vetor normal. E assim, usando desse vetor normal, pode-se gerar uma imagem 2D, ou bidimensional de todos os pontos da malha perante cada ponto. Ou seja, para cada ponto da malha, tem-se uma imagem que representa todos os demais pontos em relação a esse específico. Essa imagem é gerada calculando as distâncias perpendicular e normal de cada ponto ao ponto específico. Por ser distâncias, as coordenadas iniciais de cada ponto perante o seu sistema de coordenadas da malha são ignoradas. Então, subdividindo essa imagem de pontos por pixels, consegue-se obter uma imagem com pixels representando a quantidade de pontos em cada pixel e portanto, através de equação se consegue comparar duas imagens e gerar um coeficiente de -1 a 1 como -1 para menos correlativas e 1 para mais correlativas.

Dessa forma, comparando as imagens de cada ponto da malha A com a malha B, através de uma equação pode-se encontrar os pontos semelhantes. Os 3 pontos que derem as melhores correlações podem ser separados para o algoritmo de Horn. Como a precisão de Horn se dá ao pegar os pontos as mais distantes possível, é bom aplicar um algoritmo que calcule o tamanho da peça e defina uma distância mínima entre os três pontos escolhidos, que pode ser em torno de  $\frac{1}{4}$  do tamanho da peça.

O algoritmo de Johnson possui duas restrições perante essa tese. Uma que geometrias simétricas possuem pontos semelhantes que podem acusar uma ou mais correlatividade da imagem A em relação a imagem B.

O algoritmo de Johnson deverá ter os pontos da malha A e B bem espalhados. O número de pontos até pode ser diferente entre as malhas, pois o método calcula por intensidade de pontos em uma área da imagem, os pixels, mas a questão do espalhamento é essencial. Será discutida a solução dessa tese a seguir.

### 5.3.1.1 Problema do espalhamento de pontos

Conforme pode ser visto na Figura 5.20 e na Figura 5.21, diferentemente de um arquivo STL gerado pelo processo de escaneamento, onde os pontos tem bom espalhamento, em superfícies curvas os números de triângulos aumentam conforme a tolerância diminui ao ser gerado o arquivo do formato nativo para o formato STL. Mas ainda assim, nos meios dos triângulos não há pontos. Só haverá algum espalhamento em superfícies que variem nos 3 eixos, X, Y e Z. Em face planas é ainda pior, pois não há a necessidade de qualquer triângulo adicional.

Uma das formas de resolver esse problema, computacionalmente, é abrindo o STL, lendo a menor aresta dentre todos os triângulos e subdividindo os maiores dentre um tamanho determinando. Esse tamanho determinando pode ser a menor aresta encontrada antes de iniciar a divisão.

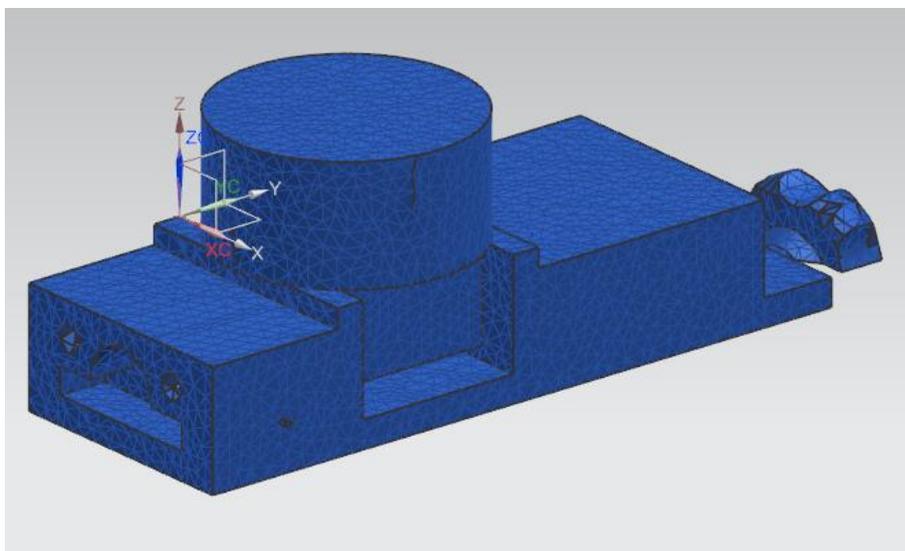


Figura 5.24: Arquivo CAD corpo de prova A salvo em STL aplicado a ferramenta de subdivisão de triângulos

Como o arquivo STL proveniente do modelamento tem que ser salvo pelo modelador, o mesmo pode usar uma função disponível em sistemas CAD que possuam as opções de engenharia reversa como o NX 9 de subdividir as facetas. Ou seja, o modelador fará o projeto da peça a ser fundida. Ao salvar essa peça, o

mesmo terá que salvar uma cópia em STL para usar na metodologia proposta por essa tese. Porém, antes de salvar, o modelador pode transformar em STL e já aplicar uma função de subdivisão das facetas deixando-as com tamanho mais uniforme.

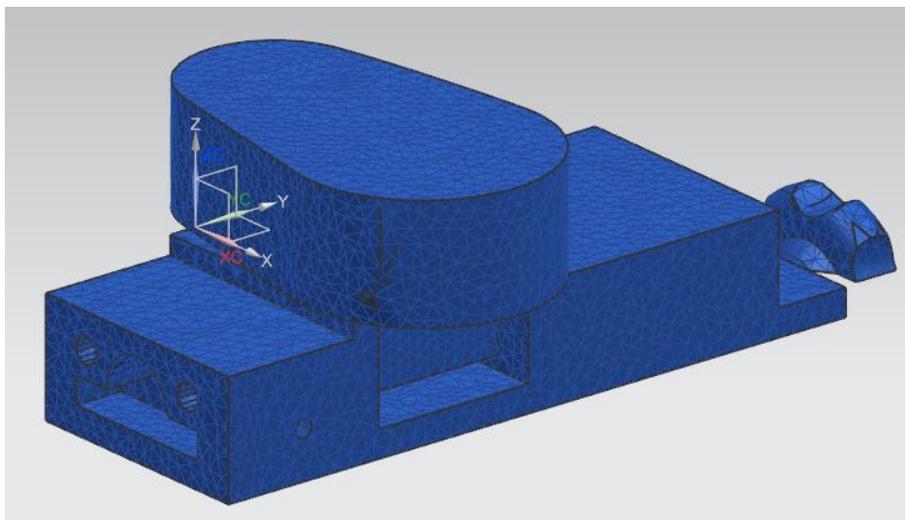


Figura 5.25: Arquivo CAD corpo de prova B salvo em STL aplicado a ferramenta de subdivisão de triângulos

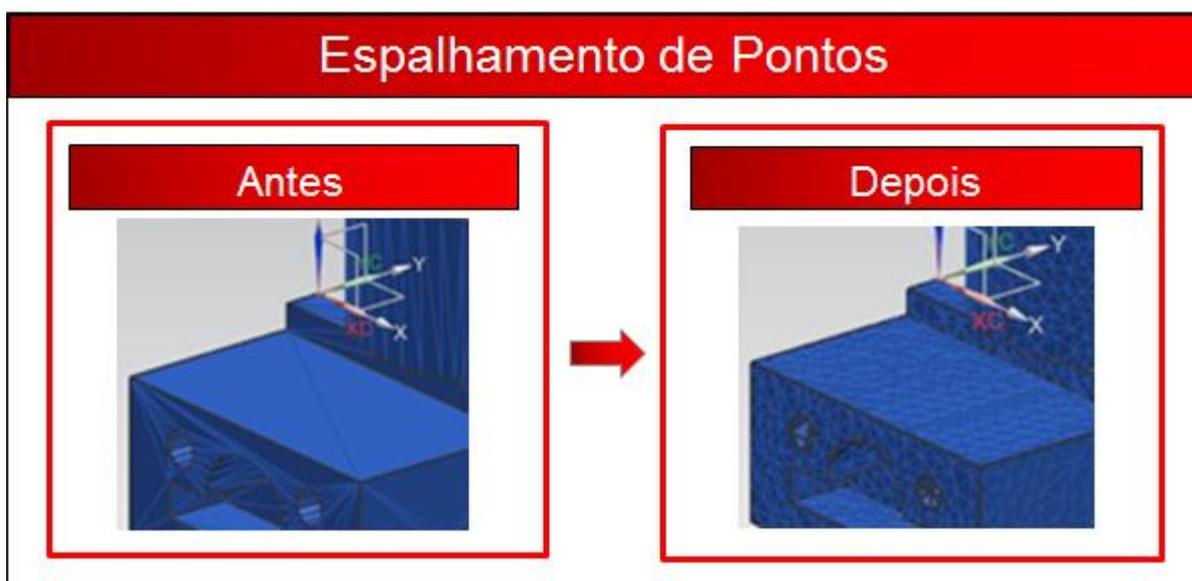


Figura 5.26: Antes e depois do comando de subdivisão das facetas ou triângulos

No caso do NX 9, existe a opção no caminho *Reverse Engineering* → *Facet Body Operations* → *Subdivide*. Dentro dessa opção há uma opção, subdividir pelo comprimento e uma opção chamada otimizar forma, que com ela todas as facetas são recalculadas. A Figura 5.24 e a Figura 5.25 ilustram os corpos de prova A e B com os pontos melhor espalhados após o uso dessa ferramenta.

A Figura 5.26 faz um comparativo do antes e o depois do uso do comando de subdivisão das facetas ou triângulos. Ou seja, ao programar essa metodologia em uma linguagem para sua execução, pode-se incluir essa matemática também no programa ou executa-la no sistema CAD antes de salvar a peça em formato STL.

### 5.3.1.2 Problema de simetria

Uma roda de alumínio injetada, como tantas outras peças do tipo, é simétrica. Essa simetria pode ocasionar um problema ao aplicar o método de Johnson. Por esse motivo, pensou-se uma forma de resolver esse problema matematicamente. Foi procurou-se uma forma de conseguir encontrar os pontos correlativos em ambas as malhas. Esses problemas nos experimentos não ocorreram, e será explicado o porquê mais adiante.

Discutindo o problema de simetria, conforme ilustrado na Figura 5.27. Na peça A, a qual se pretende fazer o alinhamento. É realizado o *SpinImage* de um ponto qualquer. Esse ponto é representado na cor azul. Então, na peça B, a qual se pretende alinhar com a peça A, encontra-se 1 ponto similar ao ponto 1 da peça A. Não importa quantos se ache igual, escolhe-se o mais similar, pois como a peça tem 4 lados iguais, serão encontrados 4 pontos semelhantes. O critério da escolha usado é o que apresentar maior índice de correlatividade.

Então, se escolhe o próximo ponto da peça A, o ponto 2. Seguindo a regra que esse ponto possua uma distância de no mínimo  $\frac{1}{4}$  do tamanho da peça. Nesse caso, conforme ilustrado, na peça A, a distância entre o ponto 1 e 2 da peça A é de 38,34 mm. Esse valor é para facilitar o entendimento do raciocínio.

Ao procurar por uma imagem semelhante na peça B, são encontrado 4 pontos. Conforme ilustra também a Figura 5.28. Fazendo um comparativo de medidas, do ponto 1 da peça B até o ponto 2A da peça B, a distância é de 38,34 mm, do ponto 2B ao ponto 1 é de 81,97 mm, do ponto 2C ao ponto 1 é de 85,19 mm e finalmente do ponto 2D ao ponto 1 é de 39,28 mm. Então, se conclui que o ponto 2 da peça B correto para se manter a proporção do sentido da peça é o ponto 2A. Pois é o que apresenta a mesma distância entre pontos da peça A.

A mesma regra se aplica ao ponto 3. Se encontra a distância na peça A e encontra o similar a peça B. Pode-se então, comparar as 3 distâncias das duas peças. A do ponto 1 ao 2, a do ponto 2 ao 3 e finalmente a do ponto 1 ao 3. Encontrando essa triangularização, os pontos podem ser inseridos no algoritmo de Horn.

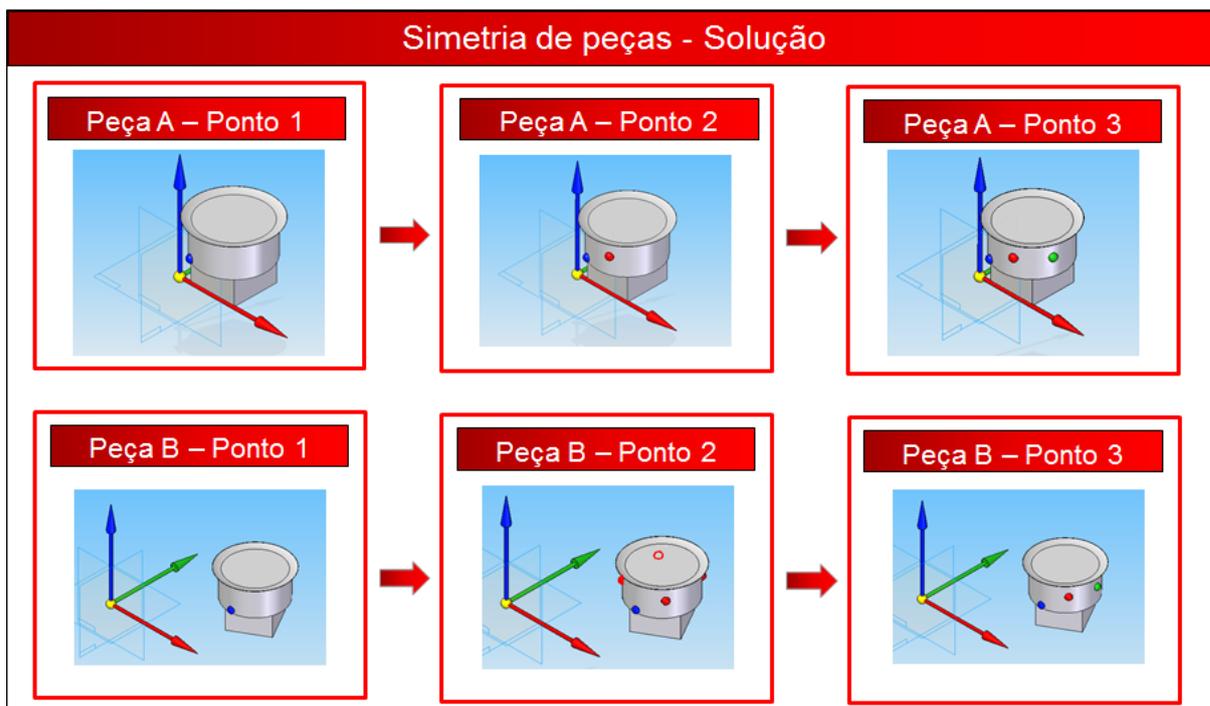


Figura 5.27: Sequência de escolha de pontos das malhas

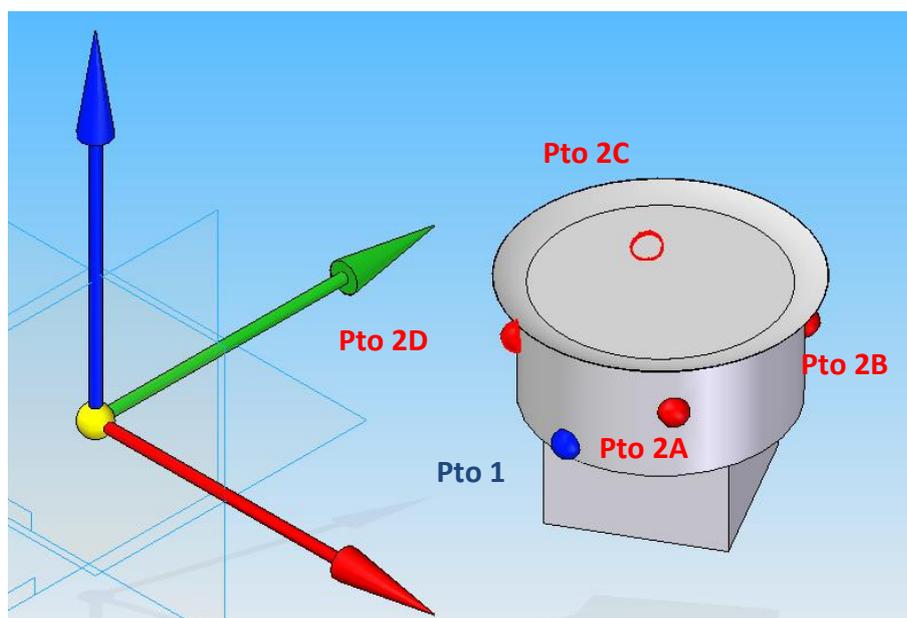


Figura 5.28: Distância dos pontos 2, em vermelho ao ponto 1, em azul

Esse método foi testado e os resultados foram apresentados na discussão acima. O método se mostrou eficaz para solucionar o problema na escolha de pontos em peças simétricas.

Porém, como o escâner não escaneia apenas o corpo de prova ou a peça, mas sim, também parte do dispositivo de fixação, esse pode também ser utilizado no

processo de alinhamento. Ou seja, como o dispositivo de fixação não é simétrico, e se for, basta colocar alguma geometria no mesmo que o deixe assimétrico, o problema de simetria deixa de ocorrer e, portanto, não há a necessidade de aplicar essa solução proposta.

Ou seja, como o dispositivo de fixação, que parte também é escaneado durante o processo, não é simétrico, o problema de simetria de geometrias não ocorre durante o processo de escaneamento.

### 5.3.2 Aplicação do Algoritmo de Horn

O algoritmo de Horn permite encontrar uma transformação que minimize os erros de coordenadas entre os pontos 1, 2 e 3 encontrados na malha A e encontrados na malha B através do *SpinImages*. Calculado essa transformação de rotação e translação, se aplica a mesma a todos os pontos da malha B para fazer alinhar o máximo possível com a malha A. Mais detalhes e explicações da transformação podem ser encontrados na dissertação de mestrado 52.

Para ilustrar como as malhas estavam desalinhadas antes do processo, a Figura 5.29 ilustra o corpo de prova modelado no sistema CAD e o arquivo escaneado pelo processo de escaneamento abertos com o auxílio de um sistema CAD, onde as duas malhas foram colocadas uma sobre a outra, demonstrando assim as diferenças do sistema de coordenadas. A Figura 5.30 ilustra o corpo de prova B.

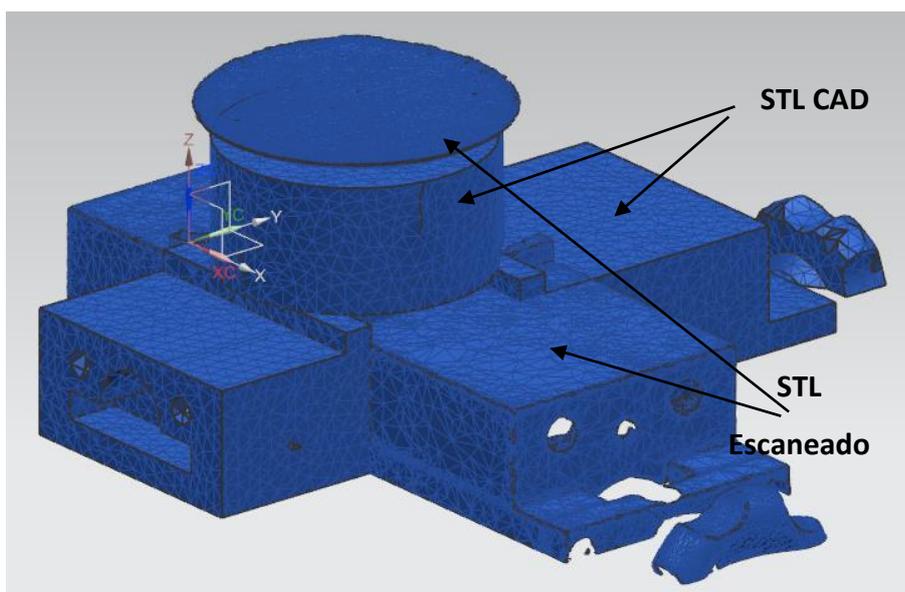


Figura 5.29: Sobreposição do arquivo CAD do corpo de prova A com o do arquivo escaneado

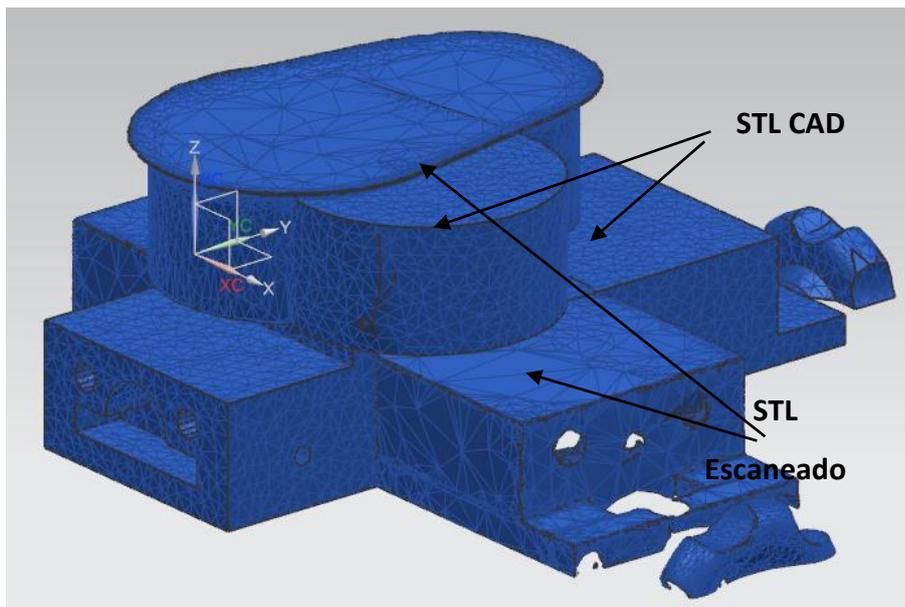


Figura 5.30: Sobreposição do arquivo CAD do corpo de prova B com o do arquivo escaneado

A Figura 5.31 ilustra o corpo de prova A após o processo de alinhamento e a Figura 5.32 ilustra o corpo de prova B depois do alinhamento.

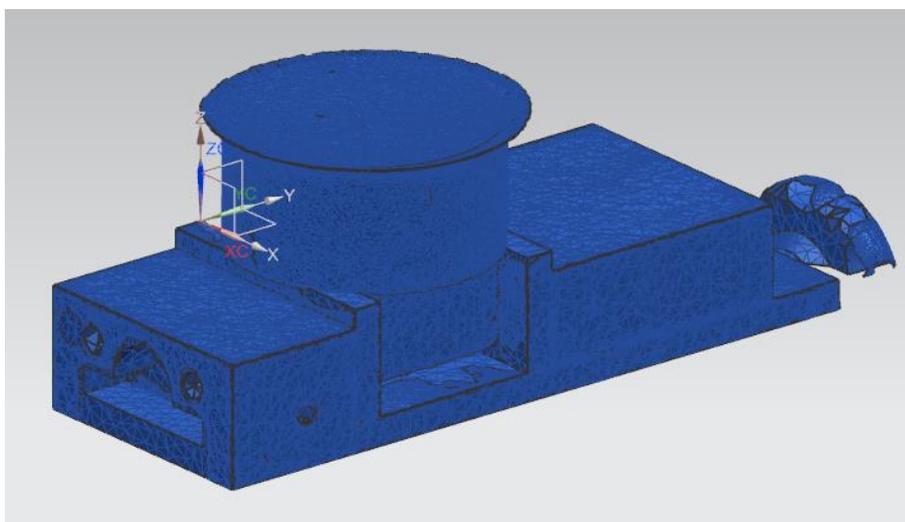


Figura 5.31: Corpo de Prova A alinhado

Após o processo de alinhamento, o arquivo STL do processo de escaneamento de ambos os corpos de prova são salvos novamente no formato STL, podendo assim ser abertos no sistema CAD para visualização dos resultados do alinhamento por Horn.

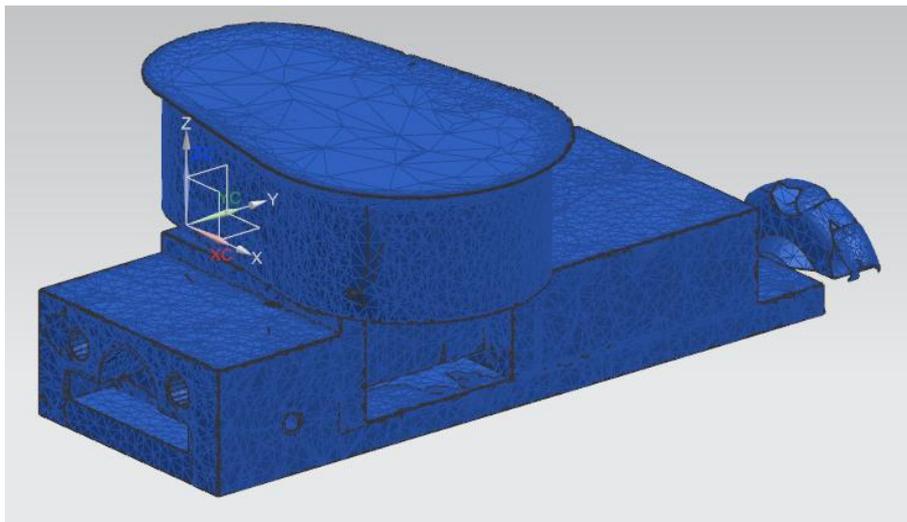


Figura 5.32: Corpo de Prova B alinhado

Os ensaios e testes também mostraram que o software que gera o arquivo STL a partir da nuvem de pontos pode facilmente alterar o sistema de coordenadas para a posição e orientação desejada. Pode-se reparar que nos arquivos dos corpos de prova A e B, os sistemas estão na mesma posição. Se essa posição já fosse a do modelo CAD, o processo de alinhamento tanto por Horn quanto por ICP poderia ser descartado, porém, com o uso de um profissional do software, o que deixaria o processo ainda com alguma influência de um operador. Em caso de um processo totalmente automatizado, o processo de alinhamento se faz necessário devido ao fato de que a matemática de transformação do arquivo de nuvem de pontos para STL não saber onde referenciar o sistema de coordenadas.

### 5.3.3 Aplicação do ICP

Como a nuvem de pontos gerada pelo processo de escaneamento possui erros e as vezes pode haver até vazios, isso é, regiões que o escâner acabou por não identificar na passada automática, essa falta de pontos ou diferença mínima na coordenadas desses pontos podem ocasionar pequenos erros no processo de geração das *SpinImages* do processo de Johnson e com isso, ocasionar pequenos erros no alinhamento por Horn.

Um exemplo de erros é que as câmeras não conseguem pegar arestas ou cantos vivos. Elas ocasionam um arredondamento. Isso porque por menor que seja a resolução dos pontos, ou seja, maior a quantidade de pontos, esses estarão muito próximos a quina, mas uns de um lado ou do outro. Quando se fecha os triângulos para fazer o arquivo STL com esses, a aresta ou canto vivo fica facetado.

O processo de ICP interage todos os pontos das duas malhas de forma a minimizar ainda mais o erro de alinhamento. Como a proposta dessa tese é apresentar formas e métodos de solução do problema, ou ainda, contribuição com o processo de automação e não entregar um software pronto e funcional, a parte do algoritmo de ICP foi realizado num software CAD manualmente. Ou seja, após a aplicação de Johnson e Horn, as malhas alinhadas foram abertas em um sistema que possui ICP e o alinhamento foi apenas refinado, comprovando assim a eficiência do uso do ICP como um refinamento do alinhamento.

#### 5.4 Processo de Reconhecimento de Rebarbas

Uma possibilidade de separar as geometrias de rebarbas, é transformar o arquivo escaneado em um sólido, então fazer uma pequena escala no mesmo, considerado tolerância, ou seja, como a superfície do processo de fundição tem erros, o processo de escaneamento tem erros e até o de alinhamento possui erros, essa tolerância é para evitar que inúmeras regiões que não são rebarbas sejam consideradas rebarbas, e então, basta usar um comando de subtração de geometrias, retirando o CAD do escaneado para separar essas geometrias.

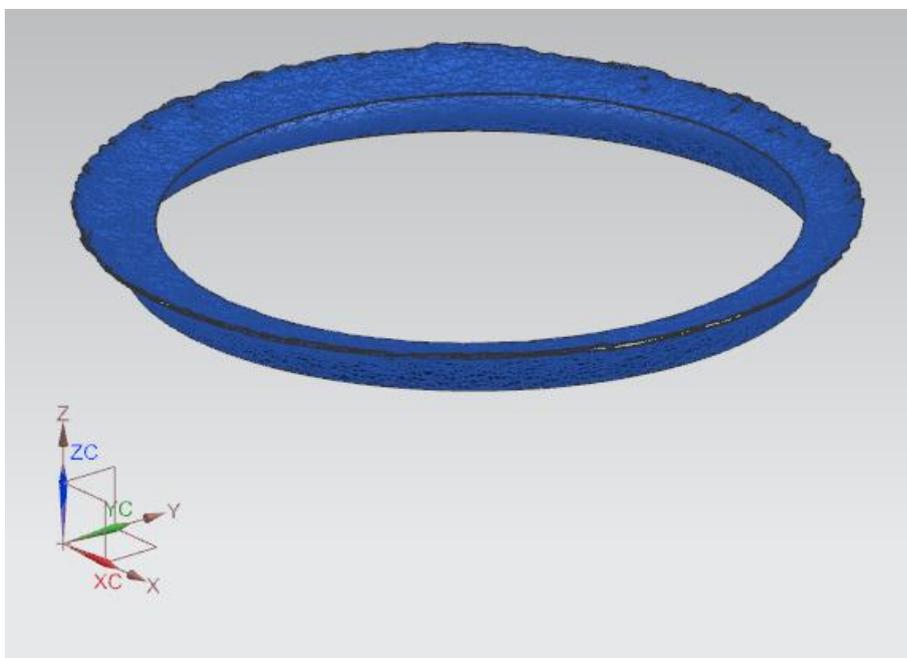


Figura 5.33: Rebarbas do corpo de prova A

Devido ao fato que se quer trabalhar apenas com triângulos, isso é, arquivos STL, a metodologia para o cálculo de rebarbas que foi aplicado é o mesmo descrito na dissertação de mestrado do autor dessa tese.

Como resultado do equacionamento, se obtém a Figura 5.33 representando a rebarba do corpo de prova A e a Figura 5.34 representado na rebarba do corpo de prova B

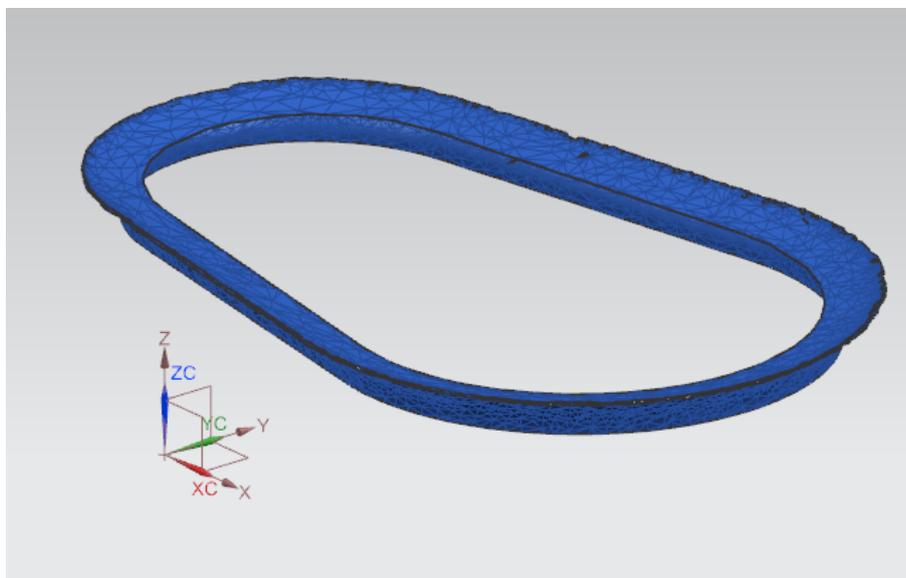


Figura 5.34: Rebarbas do corpo de prova B

## 5.5 Processo de Geração da Trajetória da Ferramenta

Através do método apresentado para reconhecer as geometrias de rebarbas, também se consegue separar do STL do arquivo CAD, a quais triângulos ou facetas cada rebarba separada se correlacionava. Ou seja, quando se encontra um ponto de uma faceta que representa uma rebarba, essa para ser encontrada foi calculada a partir de um ponto ou faceta do modelo STL da peça, basta então separar em um arquivo as facetas de rebarbas e no outro, as facetas de como deve ser a geometria sem essas rebarbas.

A parte de geração de trajetória foi realizada em um sistema CAM. No caso, o NX 9.

As rebarbas foram classificadas como geometrias do blank e as geometrias do modelo CAD, como a peça final, ou seja, onde se pretende chegar. Foi aplicada a operação *Cavity Mill* e selecionado o comando *Cut Levels* definindo a altura máxima da rebarba como o início do comando e a altura mínima da rebarba como a altura mínima do comando. E então, selecionado o comando Gerar. Isso tem a intenção de comprovar que sim, se pode programar a geração automática da trajetória da ferramenta através dos dados obtidos na etapa anterior e com o

conhecimento matemático de um sistema CAM. Ou seja, fazendo a programação similar à de um sistema CAM, pode-se obter um software que além das etapas anteriores, também faça a geração da trajetória sem a necessidade de um programador CAM.

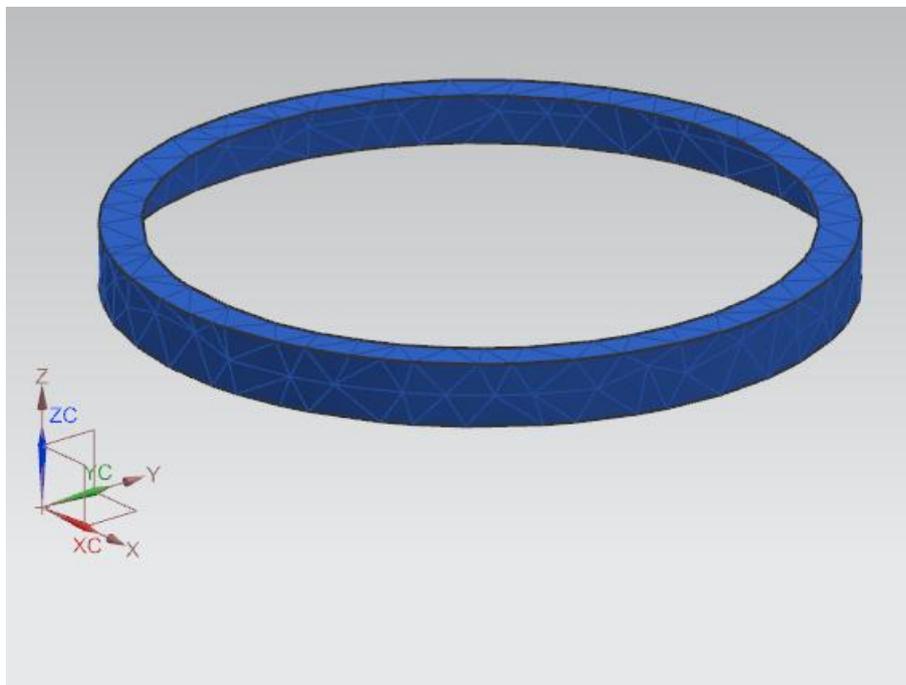


Figura 5.35: Região final após a rebarbação do corpo de prova A

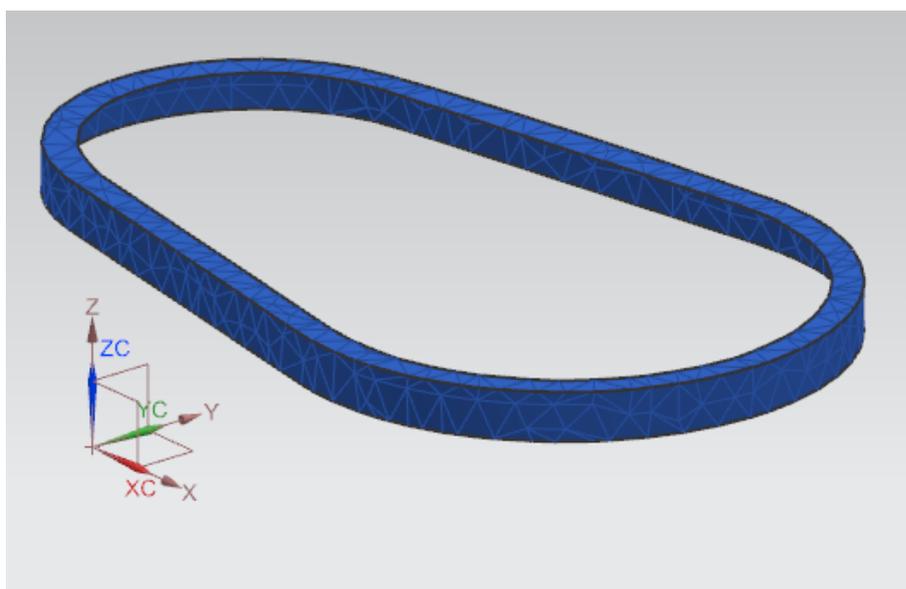


Figura 5.36: Região final após a rebarbação do corpo de prova B

O resultado do processo de separação das rebarbas já foi apresentado no capítulo anterior. A Figura 5.35 ilustra a região final do corpo de prova A Figura 5.36 e a ilustra a região final do corpo de prova B. Ou seja, essas figuras representam as facetas as quais as rebarbas encontradas no arquivo escaneado se correlacionam. Essas facetas representam a geometria que se pretende atingir depois do processo de usinagem de rebarbação.

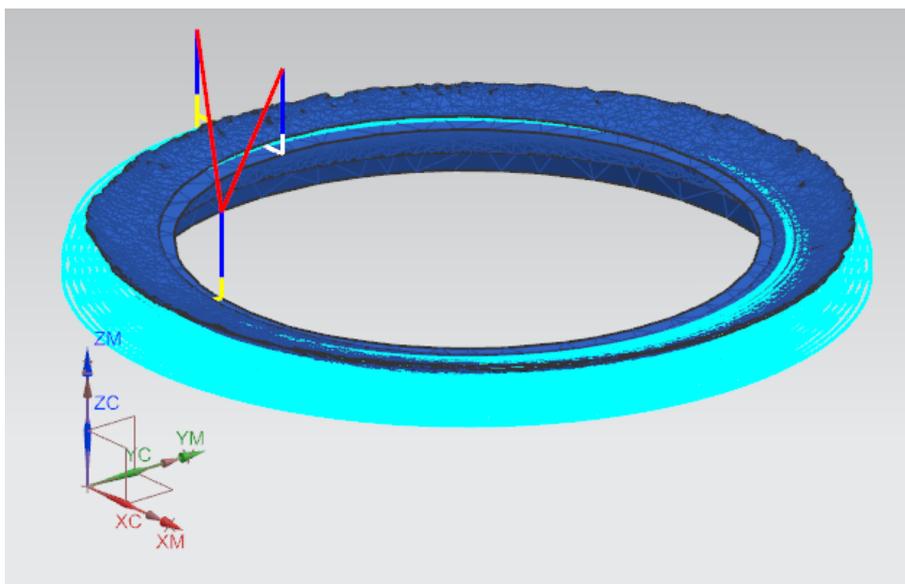


Figura 5.37: Geração da trajetória da ferramenta do corpo de prova A

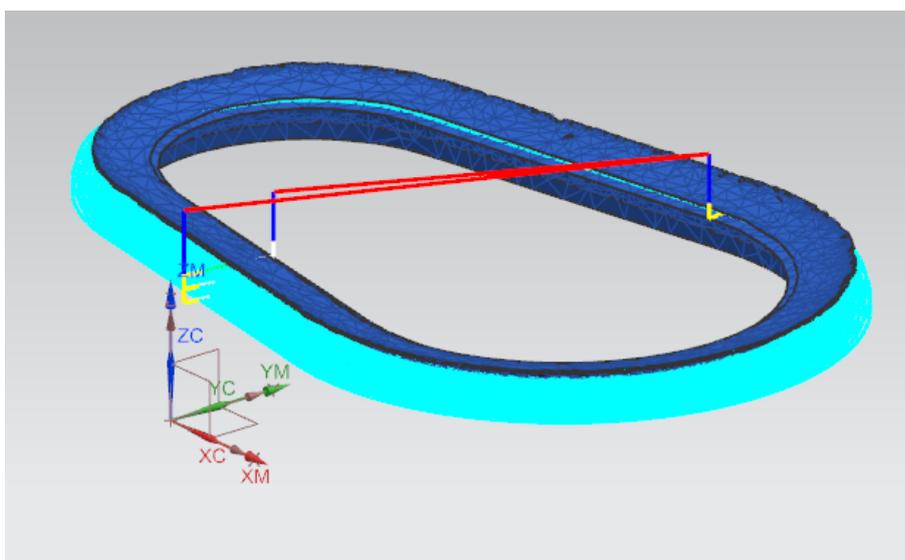


Figura 5.38: Geração da trajetória da ferramenta do corpo de prova B

Após aplicada a operação no sistema CAM, pode ser visualizada a trajetória da ferramenta. A Figura 5.37 ilustra a trajetória para o corpo de prova A e a Figura

5.38 ilustra a trajetória para o corpo de prova B. Ou seja, essas figuras demonstram a trajetória que o cabeçote fresador irá ter que fazer guiado pelo robô industrial.

### 5.5.1 Pós Processamento

O sistema CAM gerou a trajetória para a ferramenta considerando uma cinemática de 5 graus de liberdade, ou 5 eixos, como uma máquina ferramenta com X, Y e Z para translação, mais os eixos A e B para reorientar ou rotacionar a peça. O sexto eixo, pode ser considerado o eixo de rotação da ferramenta. Com isso se tem os seis graus de liberdade. Conforme visto no capítulo II, no item 2.1.3, 6 graus de liberdade são necessários para qualquer dispositivo atingir qualquer posição e qualquer orientação em seu volume de trabalho. Menos do que isso, o mecanismo não poderá alcançar todas as posições ou todas as orientações no seu volume de trabalho.

O robô utilizado nos ensaios possui 6 eixos, ou seja, 6 graus de liberdade. Ao acoplar a ferramenta de usinagem, o *spindle* como órgão terminal, é como se fosse acrescentado um sétimo eixo ao sistema.

Sistemas com mais de 6 graus de liberdade possuem mais de uma forma de atingir a mesma posição no espaço com uma determinada orientação. O problema é que é matematicamente impossível de se calcular a cinemática inversa do sistema para que através das coordenadas em X, Y e Z e a orientação, se calcular a posição angular de cada junta do robô, pois o sistema apresenta n soluções possíveis.

Há então a necessidade de transformar a trajetória gerada pelo CAM de 5 graus de liberdade para 6 graus de liberdade.

Para resolver esse problema, foi utilizado um sistema da Kuka, o CAMROB.

Esse sistema permite que o usuário defina uma orientação fixa para o órgão terminal do robô no sentido do eixo de rotação do *spindle*.

A Figura 5.39 ilustra algumas das possibilidades de posições fixas para o eixo do Spindle. Dessa forma, o sistema permite essa restrição de orientação e portanto, permite a possibilidade do pós-processamento.

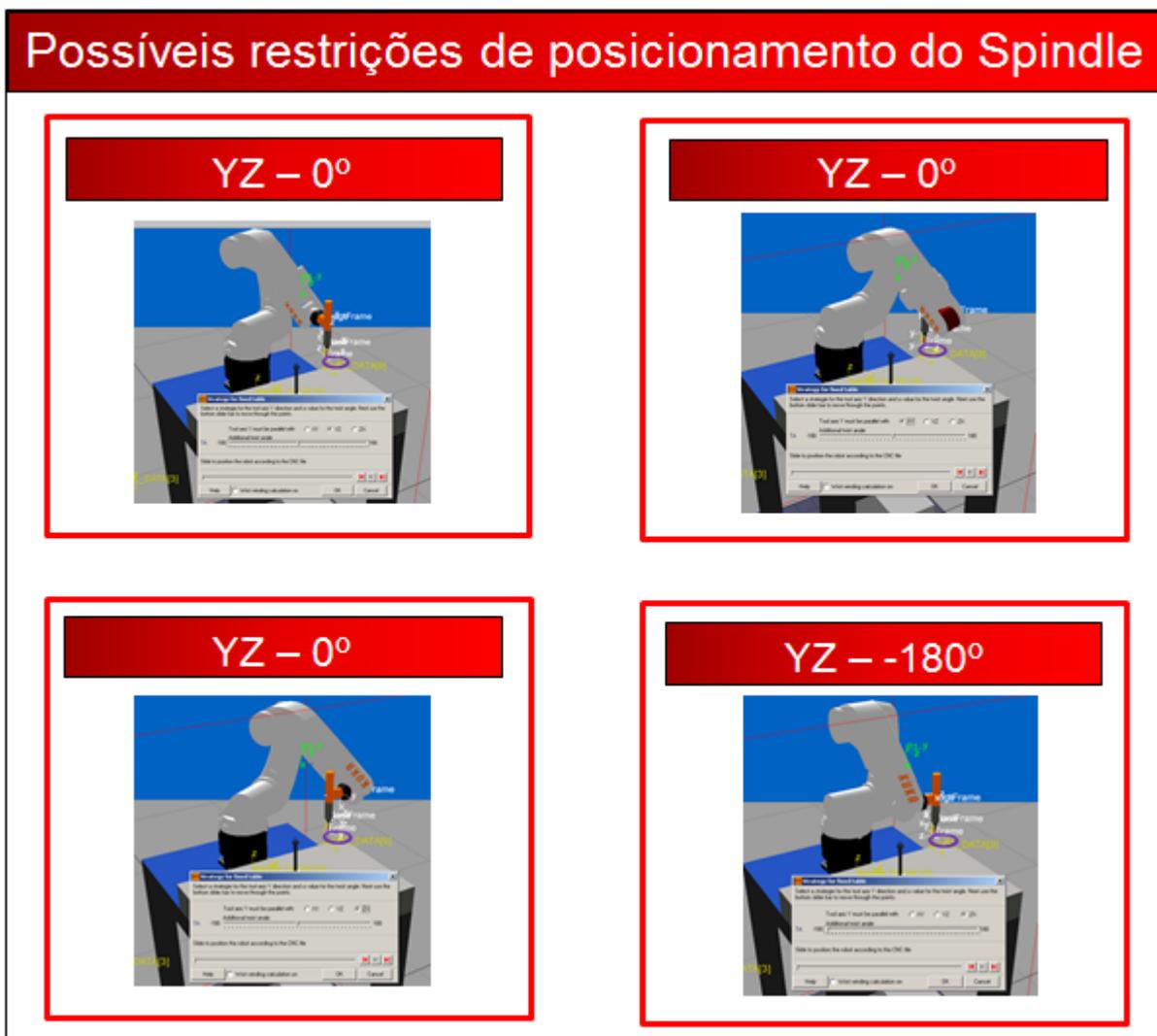


Figura 5.39: Possíveis posições para fixar a posição do Spindle

Além disso, o software permite a simulação do ambiente para verificar possíveis colisões do braço robótico com a peça ou com os dispositivos de fixação.

O procedimento de como utilizar o CAMROB passo a passo para a realização do pós-processamento está no item 8.2, o Anexo B.

### 5.5.2 Transferência dos arquivos para o comando do robô

O processo de transferência dos arquivos para o comando do robô consiste em carregar os mesmos no comando através de uma memória flash, um pen-drive, e então executar o comando de transferir.

Para transferir os arquivos do computador para o comando do robô, os seguintes passos devem ser realizados.

No comando do robô, deve-se selecionar **Monitor** → **Transfer Manage**.

Na tela que se abre, deve se selecionar a pasta do *pendrive*, para a pasta do robô D:\ApplicationsData.

Depois, deve-se ir no diretório ao qual foi copiado os arquivos, e um a um copia-los manualmente para o diretório do robô, KRC:\R1\Program. São cinco as extensões geradas pelo programa da Kuka, mas basta copiar as extensões .bin, .scr e .dat.

Ao abrir o .bin, o programa está pronto para ser executado.

## 5.6 Processo de Usinagem das Rebarbas

Nessa etapa do método proposto, os arquivos com o programa gerado para a rebarbação já está carregado no robô e a atividade de remoção das rebarbas pelo robô pode ser iniciada.

A Figura 5.40 ilustra o corpo de prova A logo após o processo de escaneamento pronto para ser rebarbado e a Figura 5.41 ilustra o corpo de prova B.



Figura 5.40: Corpo de prova A – Início da rebarbação



Figura 5.41: Corpo de prova B – Início da rebarbação

A Figura 5.42 ilustra o corpo de prova A sendo rebarbado e a Figura 5.43 ilustra o corpo de prova B.



Figura 5.42: Corpo de prova A – Processo de rebarbação



Figura 5.43: Corpo de prova B – Processo de rebarbação

Como resultado, a Figura 5.44 ilustra o corpo de prova A, com e sem rebarba. E a Figura 5.45 ilustra o corpo de prova B



Figura 5.44: Corpo de prova A, com e sem rebarba

Uma vez que todo o processo de zeramento do robô tenha sido feito de forma correta, assim como todo o procedimento de geração e pós-processamento da trajetória e também, a peça a ser rebarbada ter sido fixada no dispositivo de fixação de forma adequada, o processo de rebarbação é para ocorrer sem problemas.



Figura 5.45: Corpo de prova B, com e sem rebarba

## 5.7 Análise dos Resultados

Foram realizados quatro ensaios dos quais as dimensões finais das rebarbas variaram até no máximo um décimo de milímetro. Nas imagens apresentadas, na Figura 5.46 a geometria após a rebarbação apresentou um erro de seis centésimos e na Figura 5.47 a geometria apresentou um erro de um décimo.

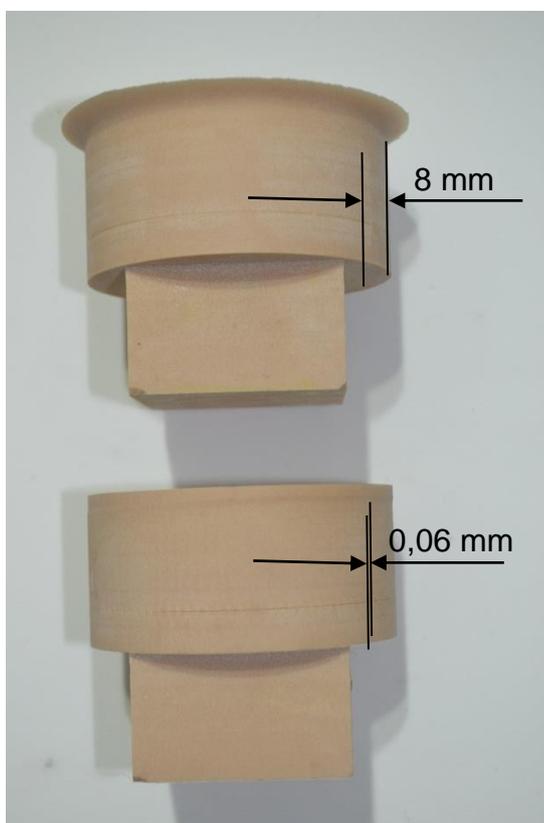


Figura 5.46: Corpo de Prova A, comparação dimensional.

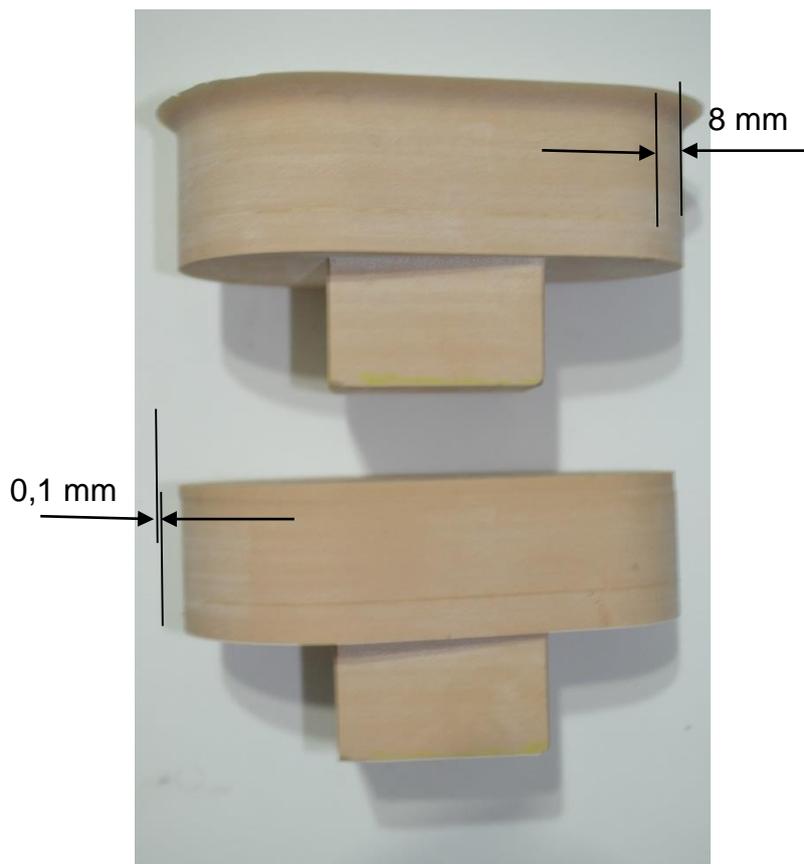


Figura 5.47: Corpo de Prova B, comparação dimensional

Os ensaios também mostraram que o maior erro se deu na direção X da base. Isso porque o dispositivo de fixação utilizado não continha um fim de curso mecânico para a peça a ser rebarbada. Ou seja, com um dispositivo ainda melhor, pode-se diminuir ainda mais essas tolerâncias de rebarbação.

## 6 Conclusões

A etapa de rebarbação durante o processo de fabricação de determinadas peças ou itens representa um gargalo na produção. Seu processo é realizado na grande maioria das vezes manualmente através de ferramentas potentes e pesadas. Por isso, é considerado insalubre e, por ser manual, não garante boa repetibilidade.

Como alternativa para a automação do processo de rebarbação, pode-se aplicar máquinas ferramentas ou robôs industriais para fazer a trajetória da ferramenta de rebarbação. Máquinas ferramentas são mais indicadas devido a rigidez mecânica, dentre outros aspectos. Porém, robôs estão cada vez mais sendo utilizados devido ao menor custo, maior volume de trabalho e flexibilidade quando comparados as máquinas ferramentas.

O processo de rebarbação exige menos do equipamento do que os processos de usinagem convencionais, isso porque o volume de material a ser removido é menor, o que faz aumentar o interesse pelo uso de robôs. Além disso, diversos estudos sobre o uso de robôs para garantir melhor posicionamento através de controle de torque, correção de trajetória dentre outros estão sendo realizados. O que torna robôs ainda mais viáveis.

Robôs já são uma realidade em processos de rebarbação e seu uso se intensifica com o passar do tempo. Mas a sua programação exige pessoal qualificado não só no processo de fabricação ao qual se aplica, mas também, no manuseio desse equipamento. Com isso, a automação da programação do mesmo se torna um requisito importante, diminuindo assim a ação do programador ou até eliminando o mesmo, melhorando o processo como um todo.

Porém, a automação não só gera benefícios. Problemas podem surgir durante a implementação ou até podem aparecer novos problemas que não havia no processo manual. Isso faz com que a metodologia para tal automação deva ser muito bem pensada e estruturada.

Estudos realizados pelo levantamento bibliográfico mostraram que o objetivo proposto por essa tese de doutorado pode ser alcançado dentro de condições de contorno com sucesso contribuindo assim para a comunidade acadêmica e para a indústria.

Diversos trabalhos e estudos mostraram que o uso de robôs já está consolidado tanto para a atividade de escaneamento, quanto para algumas das atividades de usinagem como a rebarbação.

Na questão da automação do sistema de escaneamento, estudos e experimentos práticos mostraram que o uso de mesa rotativa se mostra a alternativa mais viável para o processo de escaneamento e a mesma, contribui para melhorar a acessibilidade do robô na hora da usinagem.

O uso do escâner e do cabeçote fresador no mesmo robô se mostrou promissor e possível de ser aplicado. A proposta de um sistema de fixação para o uso tanto o escâner quanto o *spindle* separadamente torna o sistema mais duradouro evitando excesso de carga no robô e vibrações no escâner.

Quanto ao processo de alinhamento, o processo de Horn de acordo com a literatura é essencial para alinhar nuvem de pontos totalmente desorientadas, no caso, a nuvem ou malha proveniente do processo de escaneamento com a nuvem ou malha proveniente do modelo CAD da peça. Esse se mostrou funcional quando aplicado junto ao método de Johnson.

As soluções para os problemas encontrados em Johnson quanto ao espalhamento dos pontos nas malhas e quanto a questão da simetria das malhas propostas por essa tese solucionaram ambos os problemas e permitiram o uso das duas ferramentas.

O processo de alinhamento por ICP de acordo com a literatura se mostra o mais viável para o processo de refinamento do alinhamento. O mesmo pode e deve ser utilizado devido a que a nuvem escaneada possui erros discutidos nessa tese em relação ao arquivo CAD.

O processo de alinhamento é utilizado porque o escâner utilizado, caracterizado como *handscan* gera o seu zero do sistema de coordenadas aleatoriamente. Dessa forma, para se conseguir um método, e então uma programação que proporcione um sistema 100% automatizado, há a necessidade da aplicação das metodologias de alinhamento das malhas.

Porém, o software utilizado para transformar a nuvem de pontos em arquivo STL, ou seja, o software de manejo da câmera, pode através de comandos e trabalhos em cima das malhas, redefinir o sistema de coordenadas. Essa afirmação só se comprovou durante os últimos testes da tese. Ou seja, através de um retrabalho manual das nuvens no sistema comercial de uso da câmera, pode-se tornar não necessário o uso dos métodos de alinhamento das malhas, ao custo de não se obter mais um método 100% automatizado.

O processo de identificação de rebarbas, de acordo com a dissertação de mestrado [52], já apresenta resultados satisfatórios e foi novamente aplicado e comprovado no método proposto por essa tese.

Quanto o processo de geração de trajetórias, o uso de um sistema CAM comercial demonstrou e comprovou através dos experimentos a possibilidade e viabilidade de uma integração da metodologia de geração de trajetórias a metodologia de alinhamento das malhas e reconhecimento das geometrias de rebarbas. Essa sendo realizada apenas com arquivos no formato STL, comprovando assim o método ser 100% automatizado.

Os testes de geração de trajetória em um sistema CAM também demonstraram que há uma segunda possibilidade para a identificação das geometrias consideradas rebarbas e a geração da trajetória da ferramenta sobre essas. Com o uso de um programador CAD/CAM, ou seja, um método não mais 100% automatizado, esse programador pode transformar os arquivos STL em sólidos, tarefa muitas vezes não triviais e simples, mas possível de ser realizada, e então utilizar o arquivo sólido agora proveniente do escaner como sendo o blank e o modelo CAD da peça final como sendo a peça desejada, e realizar a trajetória da ferramenta a partir desses dois sólidos, não sendo mais necessário então o método e a matemática para identificação das rebarbas para a geração da trajetória.

Ou seja, o processo de identificação de rebarbas serve para que, em um sistema 100% automatizado e sem o uso de nenhum sistema comercial de CAD ou CAM, o método possa identificar as geometrias de rebarbas, ou seja, iniciais, e as geometrias finais na peça CAD, ou seja, o que se pretende atingir depois da usinagem para a posterior geração das trajetórias do cabeçote fresador guiado pelo robô.

Concluindo-se então, o processo de rebarbação utilizando robôs industriais para guiar um escaner e um cabeçote fresador pode se dar de duas formas.

Totalmente automatizado, que utiliza o método proposto por completo e todos os ensaios apresentados, ou seja, para os alinhamentos das malhas, o método de Johnson, Horn e ICP, para a identificação de rebarbas, o método proposto, desenvolvido e usado no mestrado e para a geração de trajetórias, o método utilizado em um sistema CAM implementado juntos os demais métodos.

Ou parcialmente automatizada. Que nesse caso, após a operação de escaneamento, o operador do software acerta o sistema de coordenadas do arquivo e transforma em um arquivo sólido neutro. O mesmo operador no sistema

CAM usa o arquivo CAD e o escaneado e gera as trajetórias da ferramenta. No CAMROB o mesmo pós-processa a trajetória e insere no robô.

A opção totalmente automatizada reduz a influência do programador e reduz mão de obra. Em contrapartida necessita de maior esforço para a total implementação.

A opção semi-automatizada necessita de um programador capaz de operar o software do escâner, um sistema CAM e o pós processador do robô, porém, sua implementação é mais rápida e menos custosa e possibilita a automação do processo de rebarbação em si, ou seja, a substituição do operador da esmerilhadeira ou equipamento para rebarbar pelo cabeçote fresador guiado pelo robô em um prazo de tempo bem mais curto.

De uma forma geral, a literatura mostra que automação, seja ela qual for só se torna viável para grandes números de lotes, mesmo que com poucas por lote, ou pequenos números de lotes, mas com grande quantidade de peças cada lote, ou seja, a automação só se torna viável se houver grandes quantidades de peças para serem trabalhadas.

Em contrapartida, o método apresentado por essa tese permite a automação viável de qualquer peça, mesmo que produzida em um único lote e de uma única peça. Só há a necessidade do projeto CAD da peça a ser trabalhada. Ou seja, mesmo que se projete uma única peça para ser fundida apenas uma única vez, mesmo a literatura apontando que não se compensa automatizar o processo de rebarbação da mesma, por ser um único lote de uma única peça, o método proposto por essa tese permite que ainda sim, essa peça seja rebarbada por um cabeçote fresador guiado por um robô, e não mais por um operador no chão de fábrica.

## 7 Referências

- /1/ BOGUE, R. Finishing robots: a review of technologies and applications. *Industrial Robot: An International Journal*, 2009. 6-12.
- /2/ YUNMING, Z. et al. A Study on Milling Burr Expert System in Micro-Machining. *IEEE*, 2008. 965-969.
- /3/ LIAO, L.; XI, F. (.; LIU, K. Modeling and control of automated polishing/deburring process using a dual-purpose compliant toolhead. *International Journal of Machine Tools & Manufacture*, 2008. 1454-1463.
- /4/ HSU, F.-Y.; FU, L.-C. Intelligent Robot Deburring Using Adaptive Fuzzy Hybrid Position/Force Control. *IEEE*, 2000. 325-335.
- /5/ YEO, H. S.; NGOI, B.; CHIA, L. Y. Ultrasonic Deburring. *Advanced Manufacturing Technology*, 1997. 333-341.
- /6/ BENATI, F. et al. A robot-based burr measurement system for the automotive industry. *IEEE*, 1999. 1740-1744.
- /7/ ABELE, E.; KULOK, M.; WEIGOLD, M. Machining with industrial Robots: analysis of the system behavior. *Ciência e Tecnologia*, p. 72, 2006.
- /8/ PAN, Z.; ZHANG, H. Robotic machining from programming to process control: a complete solution by force control. *Emerald*, 2008. 400-409.
- /9/ JAYAWEERA, N.; WEBB, P. Robotic edge profiling of complex components. *Emerald*, 2011. 38-47.

- /10/ CHEN, S.-C.; TUNG, P.-C. Trajectory planning for automated robotic deburring on an unknown contour. *Machining Tools e Manufacture*, 2000. 957-978.
- /11/ KOCHAN, A. Deburring: a joint operation. *Assembly Automation*, 1996. 29-30.
- /12/ KOHRT, C. et al. An online robot trajectory planning and programming support system for industrial use. *Elsevier*, 3 Setembro 2014. 71-79.
- /13/ BURBIDGE, R.; WILSON, M. S. Vector-valued function estimation by grammatical evolution for autonomous robot control. *Elsevier*, 22 Setembro 2013. 182-199.
- /14/ YANG, X. et al. A minimal kinematic model for serial robot calibration using POE formula. *Elsevier*, 13 Novembro 2013. 326-334.
- /15/ ABREU, P. Robótica Industrial. *Mestrado em Automação, Instrumentação e Controle*, 2002. 29.
- /16/ ROMANO, V. F. *Robótica Industrial*. Sao Paulo: Edgard Blücher, 2002.
- /17/ VUCINA, D.; CURKOVIC, M.; NOVKOVIC, T. Classification of 3D shape deviation using feature recognition operating on parameterization control points. *Computers in Industry*, 2014. 1018–1031.
- /18/ ASÍS, L. F. D. et al. A statistical method for geometry inspection from point clouds. *Applied Mathematics and Computation*, 2014. 562–568.
- /19/ NIKU, S. B. *Introdução à Robótica*. Rio de Janeiro: LTC, 2011.

- /20/ ROSÁRIO, J. M. *Princípios de Mecatrônica*. Sao Paulo: Pearson, 2005.
- /21/ GROOVER, M. P. et al. *Industrial Robotics*. United States of America: McGraw-Hill, 1939.
- /22/ CRAIG; J., J. *introduction to Robotics - Mechanics and Control*. New Jersey: Pearson, 2005.
- /23/ INTERNATIONAL FEDERATION OF ROBOTICS. *Robots for Competitive Industries*. Brisbane: MEP, 1993.
- /24/ KIM, C.; TO, J. H. C. Application of an active pneumatic actuator to robotic deburring. *Emerald*, 2007. 487-494.
- /25/ ABB. IRB 6620 - The agile spot welder. Disponível em: <<http://new.abb.com/products/robotics/industrial-robots/irb-6620>>. Acesso em: 15 Outubro 2014.
- /26/ KUKA. KR 6-2. Disponível em: <[http://www.kuka-robotics.com/br/products/industrial\\_robots/low/kr6\\_2/](http://www.kuka-robotics.com/br/products/industrial_robots/low/kr6_2/)>. Acesso em: 15 Outubro 2014.
- /27/ II, E. J. L.; BRACARENSE, A. Q. Trajectory generation in robotic shielded metal arc welding during execution time. *Emerald*, 2009. 19-26.
- /28/ CHEN, X. Z.; CHEN, S. B. The autonomous detection and guiding of start welding position for arc welding robot. *Emerald*, 2010. 70-78.

- /29/ MIN, S.; KIM, J.; DORNFELD, D. A. Development of a drilling burr control chart for low alloy steel, AISI 4118. *Elsevier*, 2001. 4-9.
- /30/ ABB. IRB 1600ID - Dedicated arc welding robot. Disponível em: <<http://new.abb.com/products/robotics/industrial-robots/irb-1600id>>. Acesso em: 15 Outubro 201.
- /31/ HEPING, C.; THOMAS, F.; XIONGZI, L. A review of CAD-based robot path planning for spray painting. *Emerald*, 2009. 45-50.
- /32/ ABB. IRB 52 - A compact painting specialist. Disponível em: <<http://new.abb.com/products/robotics/industrial-robots/irb-52>>. Acesso em: 15 Outubro 2014.
- /33/ KIM, J. et al. A practical approach for minimum-time trajectory planning for industrial robots. *Emerald*, 2010. 51-61.
- /34/ SCHNEIDER, U. et al. Compensation of Errors in Robot Machining With a Parallel 3D-Piezo Compensation Mechanism. *Forty Sixth CIRP Conference on Manufacturing Systems 2013*, 2013. 305-310.
- /35/ PIRES, J. N. et al. Force/Torque sensing applied to industrial robotic deburring. *Emerald*, 2002. 232-241.
- /36/ FARZAD, R. et al. Angular analysis of the cyclic impacting oscillations in a robotic grinding process. *Elsevier*, 05 Julho 2013. 160-176.
- /37/ ABB. IRB 6660 - The optimised pre-machining robot. Disponível em: <<http://new.abb.com/products/robotics/industrial-robots/irb-6660-for-pre-machining>>. Acesso em: 15 Outubro 2014.

- /38/ OLSSON, T. et al. Cost-efficient drilling using industrial robots with high-bandwidth force feedback. *Elsevier*, 30 Janeiro 2009. 24-38.
- /39/ UHLMANN, E. et al. Applicability of industrial robots for machining and repair processes. *2nd International Through-life Engineering Services Conference*, 2013. 234-238.
- /40/ ZILIANI, G.; VISIOLI, A.; LEGNANI, G. A mechatronic approach for robotic deburring. *Mechatronics*, 2007. 431-441.
- /41/ NAGATA, F. et al. Development of CAM system based on industrial robotic servo controller without using robot language. *Robotics and Computer-Integrated Manufacturing*, 12 Outubro 2012. 454-462.
- /42/ A., A. H. et al. Towards fully automatic reliable 3D acquisition: From designing. *Robotics and Autonomous Systems*, 31 Marco 2014. 1197-1207.
- /43/ KARASZEWSKI, M.; SITNIK, R.; BUNSCH, E. On-line, collision-free positioning of a scanner during fully automated three-dimensional measurement of cultural heritage objects. *Robotics and Autonomous Systems*, 14 Março 2012. 1205-1219.
- /44/ LI, J. et al. Calibration of a multiple axes 3-D laser scanning system consisting of robot. *Optik*, 05 Fevereiro 2010. 324–329.
- /45/ AHMADABADIAN, A. H. et al. A comparison of dense matching algorithms for scaled surface reconstruction using stereo camera rigs. *ISPRS Journal of Photogrammetry and Remote Sensing*, 08 Março 2013. 157–167.

- /46/ AHMADABADIAN, A. H. et al. Image Selection in Photogrammetric Multi-View Stereo Methods for Metric and Complete 3D Reconstruction. *Proc. of SPIE Vol. 8791*, 2013.
- /47/ CHOWDARY, B. V. et al. An investigation for improvement of the 3D-digitization process: a reverse engineering approach. *Journal of Manufacturing Technology Management*, Julho 2010. 131-147.
- /48/ LARSSON, S.; KJELLANDER, J. A. P. Path planning for laser scanning with an industrial robot. *Robots and Autonomous Systems*, 2008. 615-624.
- /49/ NIOLA, V. et al. A method for the calibration of a 3-D laserscanner. *Robotics and Computer-Integrated Manufacturing*, 24 Junho 2010. 479–484.
- /50/ KARASZEWSKI, M.; SITNIK, R.; BUNSCH, E. On-line, collision-free positioning of a scanner during fully automated three-dimensional measurement of cultural heritage objects. *Robotics and Autonomous Systems*, 13 Março 2012. 1205-1219.
- /51/ JOHNSON, A. E. S.-I. *A Representation for 3-D Surface Matching - Doutorado*. Pittsburgh: Carnegie Mellon University, 1997.
- /52/ OLIVEIRA, B. B. D. *Identificação Automática de Rebarbas em Peças Fundidas: Uma Contribuição para a Automação do Processo de Rebarbação*. Santa Bárbara D'Oeste. 2011.
- /53/ HORN, B. K. P. Closed-form solution of absolute orientation using unit quaternions. *Journal Of The Optical Society Of America*, Abril 1987. 629-6.

- /54/ VIEIRA, T. M. D. A. *Registro Automático de Superfícies Usando Spin-Image - Mestrado*. Universidade Federal de Alagoas: [s.n.], 2007.
- /55/ BAUER, J. *Methoden der Offline-Bahnkorrektur für die spanende Bearbeitung mit Industrierobotern*. TU Darmstadt. Darmstadt, p. 153. 2011.
- /56/ XIAO, W.; HUAN, J.; DONG, S. A STEP-compliant Industrial Robot Data Model for robot off-line programming systems. *Elsevier*, 12 Setembro 2013. 114-123.
- /57/ ZENG, W.; WANG, C. Learning from NN output feed back control of robot manipulators. *Elsevier*, 15 Marco 2013. 172-182.
- /58/ DU, G.; PINGZHANG. Marker less human – robot interface for dual robot manipulators using Kinect sensor. *Elsevier*, 18 Outubro 2013. 150-159.
- /59/ OLIVEIRA, L. et al. RSSI-based relative localisation for mobile robots. *Elsevier*, 31 Agosto 2013. 321-335.
- /60/ STEWART, J. *Cálculo Volume II*. Sao Paulo: Pioneira, 2001.
- /61/ CALLIOLI, C. A.; DOMINGUES, H. H.; COSTA, R. C. F. *Álgebra Linear e Aplicacoes*. Sao Paulo: Atual, 2011.
- /62/ BOLDRINI, C.; FIGUEIREDO, W. *Álgebra Liner*. Sao Paulo: Hembra, 1980.
- /63/ BAJPAI, A. C.; MUSTOE, L. R.; WALKER, D. *Matemática para Engenheiros*. Sao Paulo: Hemus, 1980.

/64/ WIKIPEDIA ENCYCLOPEDIA. *Wikipedia*, 01 Agosto 2014. Disponível em:  
<[http://en.wikipedia.org/wiki/Cubic\\_function](http://en.wikipedia.org/wiki/Cubic_function)>.

/65/ WIKIPEDIA ENCYCLOPEDIA. *Wikipedia*, 05 Agosto 2014. Disponível em:  
<[http://en.wikipedia.org/wiki/Quartic\\_function](http://en.wikipedia.org/wiki/Quartic_function)>.

/66/ ANDREIS, D.; CANUTO, E. S. Orbit dynamics and kinematics with hll  
quaternions. *American Control Conference*, 2 Julho 2004. 3660-366.

/67/ MORTENSON, M. E. *Geometric Modeling*. USA: Braun-Brumfield, 1985.

/68/ ANDERL, R. *Grundlagen des CAE/CAD*. Darmstadt: [s.n.], 2009.

/69/ JOHNSON, A. E. *Spin-Images: A Representation for 3-D Surface Matching* -  
Tese Doutorado. Pittsburgh: Carnegie Mellon University, 1997.

/70/ PERES, F. O. *SCANNER 3D: PROBLEMAS E SOLUÇÕES*. LONDRINA-PR.  
2013.

## 8 Anexos

### 8.1 Anexo A – Masterização do robô Kuka KR5

Esse anexo tem como função auxiliar na masterização de um ou mais eixos do robô kuka KR5.

#### **Procedimento:**

- Posicionar cada eixo do robô para as suas devidas posições de masterização.
- Masterizar os eixos através do comando do robô.

#### **Execução:**

Antes de posicionar os eixos do robô, o mesmo deve ser movido para a posição zero e a tampa que cobre o motor 3 deve ser removida, conforme a Figura 8.1. A movimentação dos eixos deve ser feita no nível junta a junta, ou seja, cada eixo movimentado isoladamente. A posição zero ilustrada na Figura 8.1, pode ser aproximada. Isso porque essa posição é apenas uma posição de pré-referência para então posicionar cada eixo em sua devida posição de masterização.

Para realizar as movimentações no robô, deve-se atentar aos tipos de movimentos que o robô pode fazer.

O robô pode atuar em quatro modos, conforme a Figura 8.2:

1. Manual com velocidade reduzida. T1.
2. Manual com velocidade alta. T2.
3. Automatico. Aut.
4. Automático Externo. Aut Ext.



Figura 8.1: Posição inicial de masterização.

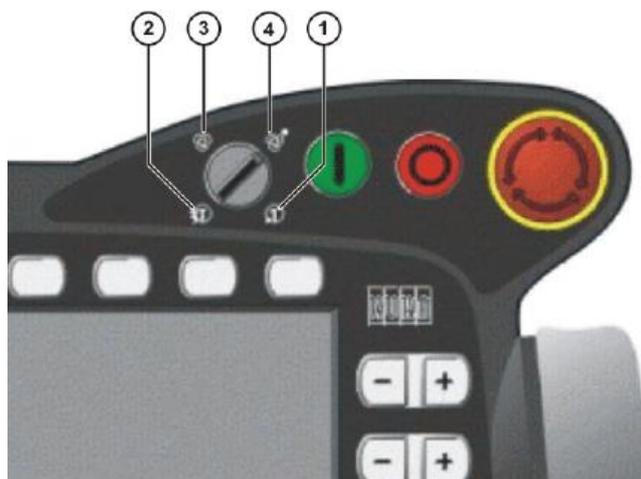


Figura 8.2: Modos de Operação do Robô.

Os modos devem usados da seguinte forma:

1. T1: Para programação e teste de operação. A velocidade é reduzida para evitar possíveis colisões.
2. T2: Para teste de operação. A velocidade é a real programada.
3. Aut: Para operação constante. A velocidade é a programada.
4. Aut Ext: Para operação constante comandada por um circuito externo. A velocidade é a programada.

Para a movimentação de masterização deve-se usar o modo T1.

Deve se agora observar no canto superior esquerdo do comando o indicativo de não movimento dos eixos conforme a Figura 8.3

Indicativo de não movimento dos eixos

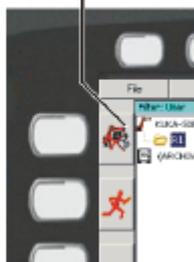


Figura 8.3: Estado da movimentação dos eixos

O indicativo atual significa que nenhum dos eixos estão habilitados a se movimentar. Esse estado pode ser mudado para duas formas diferentes, movimentação pelas Jog Keys  ou pelo Space Mouse .

Após selecionar a preferência de entrada de movimento para o comando. Deve-se escolher dentre as opções eixo-a-eixo , coordenadas mundo, de ferramenta ou de base.

Como o robô nessa etapa não está masterizado, só se pode movimentar na opção junta a junta.

Lembrando que as diferenças são que junta a junta, cada eixo é movimentado isoladamente. Coordenadas mundo, a extremidade do robô se move de acordo com a orientação padrão do robô que está na base do robô virtualmente. Coordenadas ferramenta, a extremidade do robô se move com a orientação calibrada para ferramenta a ser utilizada e coordenadas de base, a extremidade do robô se move em relação a base ou dispositivo de fixação zerado nesse tipo de coordenada.

Para conseguir mover os eixos no moto T1, um dos três botões de *Enabling* devem estar pressionados. A Figura 8.4 mostra a localização dos mesmos. Ao serem pressionados, pode-se notar que eles possuem três posições. São elas solto, pressionado no nível intermediário e pressionado completamente. Deve-se operar no nível intermediário. Isso se deve a questão de segurança. Em caso de acidente, a reação do operador pode ser soltar o botão ou pressionar ele mais forte. Portanto, o botão saindo da posição intermediária o robô para.



Figura 8.4: Botões de Enable

Estando o robô configurado para movimentar junta a junta, o botão *enable* pressionado ao nível intermediário, movimentar o eixo 1 de acordo com a Figura 8.5

O robô deve estar a aproximadamente 5 mm do parafuso colocado para a masterização conforme a Figura 8.6.



Figura 8.5: Posicionamento do eixo 1 do robô. Figura A

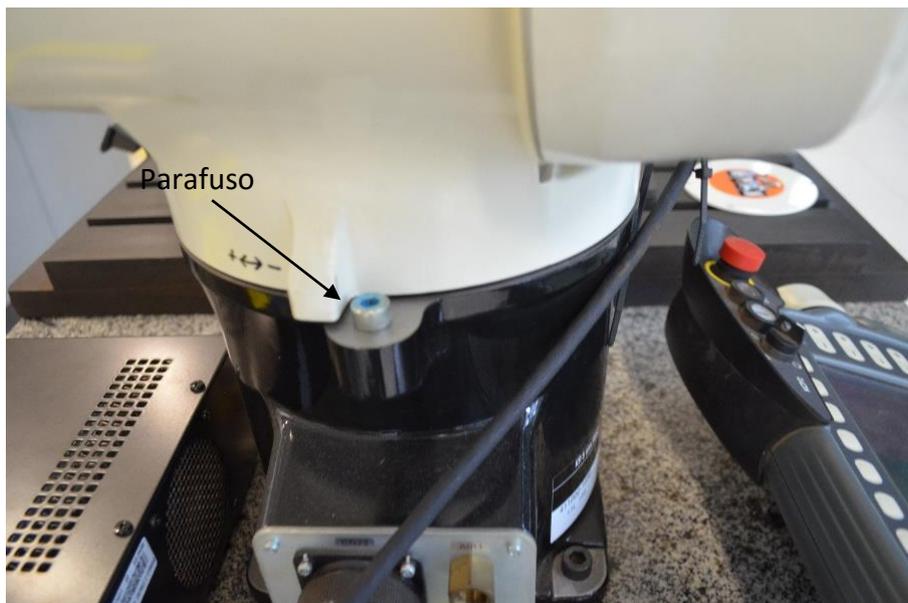


Figura 8.6: Posicionamento do eixo 1 do robô. Figura B

O eixo 2 deve ser movido para a posição ilustrada na Figura 8.7. Conforme a Figura 8.8, o parafuso deve ficar a aproximadamente 270 mm da base do robô.



Figura 8.7: Posicionamento do eixo 2 do robô. Figura A



Figura 8.8: Posicionamento do eixo 2 do robô. Figura B

A tampa do motor 3 aberta agora está do outro lado da mesa. Deve-se observar que há um parafuso cinza maior. Deve-se então remover esse parafuso e separá-lo, pois será usado para masterizar o eixo 4 posteriormente. A Figura 8.9 ilustra o lugar onde é encontrado o parafuso e a Figura 8.10 ilustra qual deles deve ser removido.

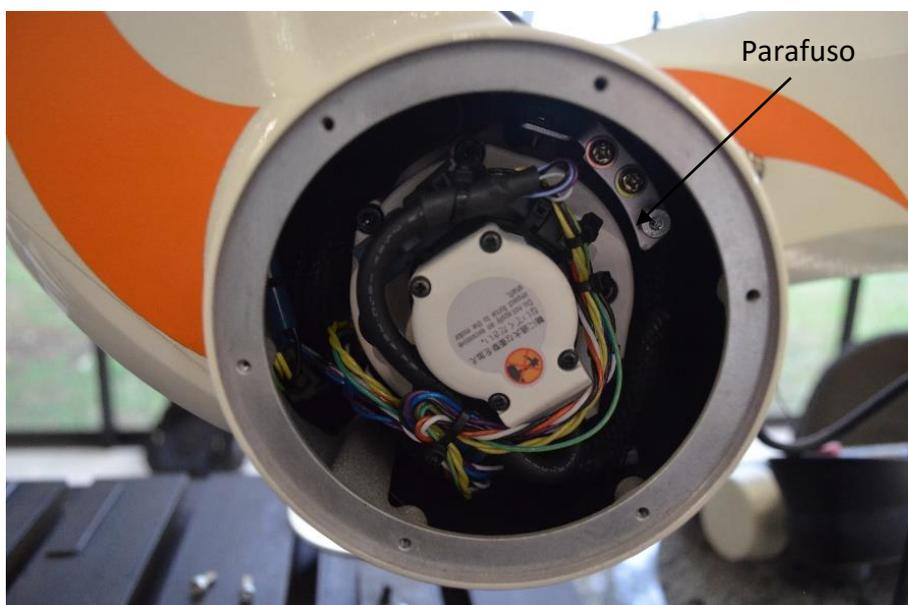


Figura 8.9: Ilustração da localização do parafuso para masterizar o eixo 4.



Figura 8.10: Ilustração da localização do parafuso para masterizar o eixo 4, parafuso removido.

O eixo 4 deve ser posicionado antes do eixo 3. Para tal o manual da Kuka ressalta a importância de observar o cabo preto que se enrola entorno do motor 3. Ele deve estar conforme a Figura 8.11. Se esse cabo estiver enrolado dentro do robô, esse se romperá. Portanto, ao movimentar o eixo 4 para a posição de masterização, deve-se mover o eixo devagar e deve-se observar cuidadosamente o cabo preto. Se esse começar a ficar na forma errada, deve-se movimentar o eixo na direção oposta.

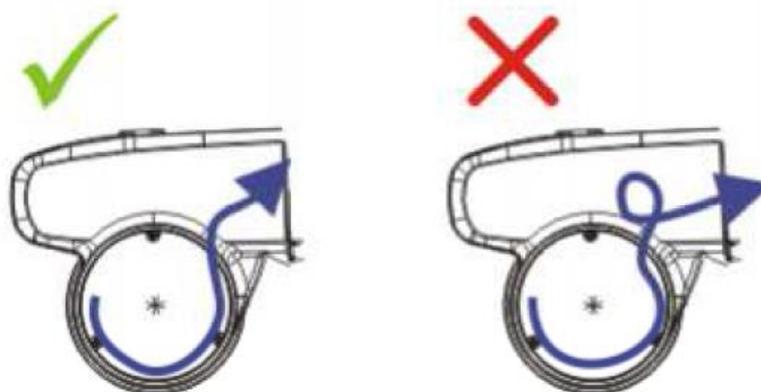


Figura 8.11: Jeito correto do cabo preto.

Tendo a observação anterior ressaltada, o eixo 4 deve-se então ser movimentado no sentido negativo do eixo de forma a ficar igual a Figura 8.12 e então, deve-se colocar o parafuso retirado de dentro do lugar onde fica o motor 3. Então, o eixo 4 deve ser alinhado conforme a Figura 8.13. Deve haver uma distância de aproximadamente 2 mm entre o parafuso e o corpo do robô.



Figura 8.12: Posicionamento do eixo 4 para a colocação do parafuso de masterização.



Figura 8.13: Posicionamento do eixo 4 após a colocação do parafuso de masterização.

O eixo 3 deve ser posicionado de forma que o parafuso mostrado na Figura 8.14 até a tangente do robô fique a uma distância aproximada de 135 mm.



Figura 8.14: Posicionamento do eixo 3.

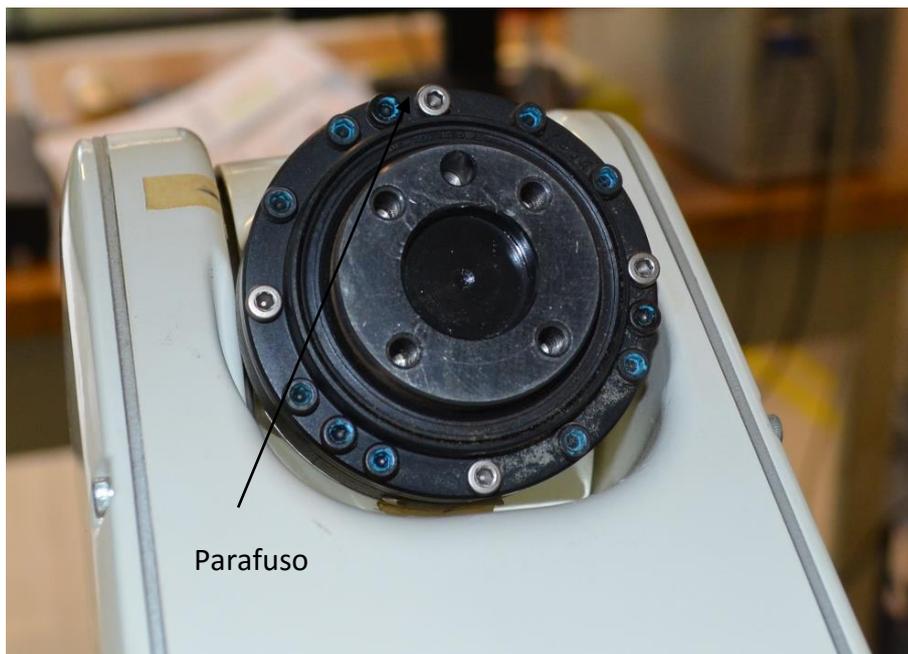
O eixo 5 deve-se então ser posicionado conforme a Figura 8.15. Deve ficar a uma distância de aproximadamente 5 mm com a tangente do robô.



Figura 8.15: Posicionamento do eixo 5.

Para masterizar o eixo 6, deve se montar a peça de masterização na flange do robô.

Mas antes, a mesma deve estar posicionada conforme a Figura 8.16. O furo do pino guia da flange deve estar para cima, alinhado o mais próximo possível com o parafuso cinza do robô.



Parafuso

Figura 8.16: Posicionamento do eixo 6 para a montagem da peça de masterização.

Então deve-se montar a peça de masterização conforme a Figura 8.17.



Figura 8.17: Montagem da peça de masterização.

O robô antes de ser masterizado deve estar conforme a Figura 8.18.



Figura 8.18: Posicionamento dos eixos para a masterização

Feito isso, no controle do robô, deve ser selecionada a função **Setup** → **Master** → **Set** → **Mastering**. Então deve-se pressionar a tecla *Master*. Deve-se então pressionar um dos três botões de *Enable* e então a tecla verde *Start*.

O robô deverá se mover automaticamente tocando ou nos parafusos de masterização, ou nas tangentes de seu corpo ou ainda na peça de masterização até que todos os eixos estejam masterizados. Pode haver ou não a necessidade de entrar novamente na opção de masterização após a masterização de algum dos eixos. Mas os experimentos mostraram que esse tipo de ocorrência não afeta a masterização.

## 8.2 Anexo B - Pós-Processamento Utilizando CamRob

Nesse anexo segue o método de se realizar o pós processamento utilizado nessa tese. Está exemplificado para o corpo de prova A, mas o mesmo se repete para o corpo de prova B.

Após gerar a trajetória da ferramenta no sistema CAM, deve-se extrair o arquivo CLSF. No caso no NX, através do comando **Output CLSF**.

Então, no sistema Kuka SimPro, deve-se modelar o ambiente de trabalho e colocar o robô em sua posição o mais correta possível. A Figura 8.19 ilustra o ambiente de trabalho modelado no sistema.

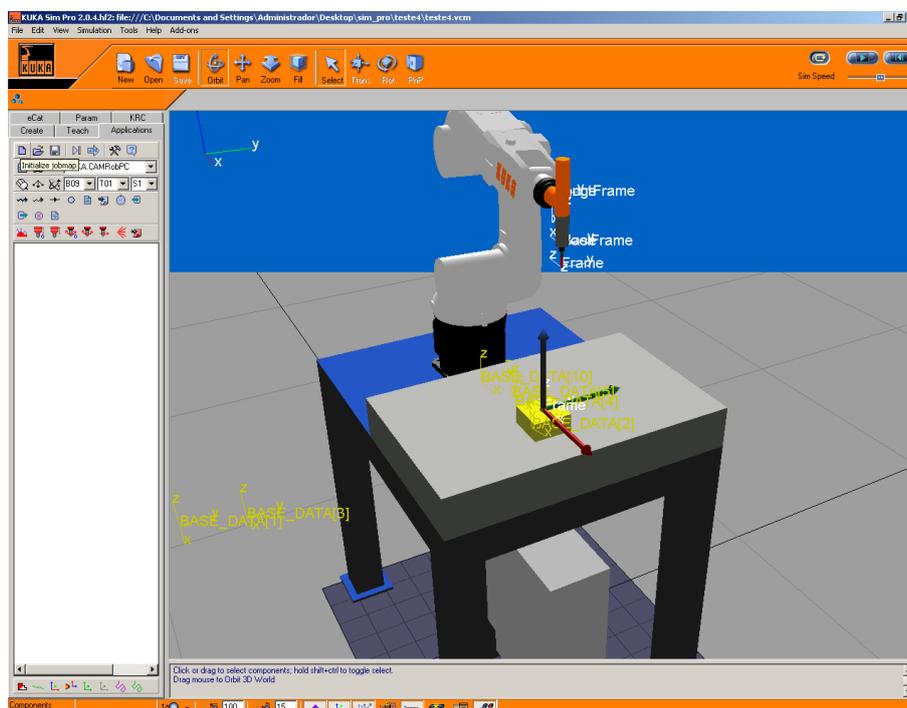


Figura 8.19: Tela inicial do Kuka SimPro para início do pós processamento.

Então, deve-se iniciar um novo trabalho através do comando Novo conforme a Figura 8.20. Deve-se alterar os 3 primeiros campos, que são diretório que se deseja salvar, nome do projeto e nome do trabalho, cujo o último é o que vai ser inserido no comando do robô. Os demais campos são de menor importância e não são necessários para a funcionalidade de pós-processamento.

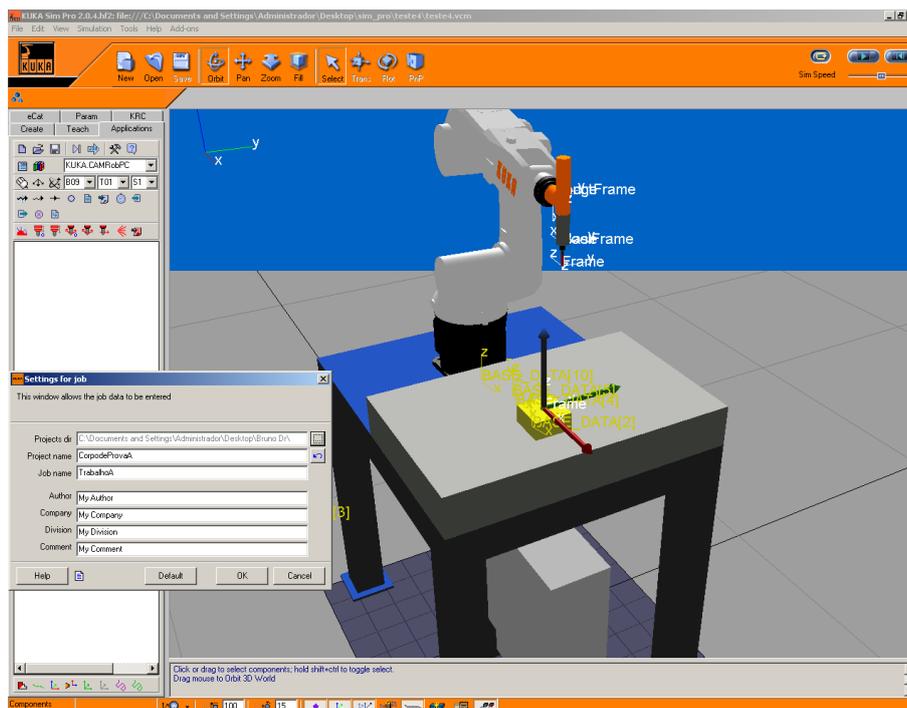


Figura 8.20: Novo trabalho

Agora, deve-se adicionar o processo, ou seja, o arquivo CLSF gerado no sistema CAM.

Deve-se selecionar o comando **Add Process Comand**, conforme indicado na Figura 8.21 e selecionar na janela que se abre, o arquivo desejado.

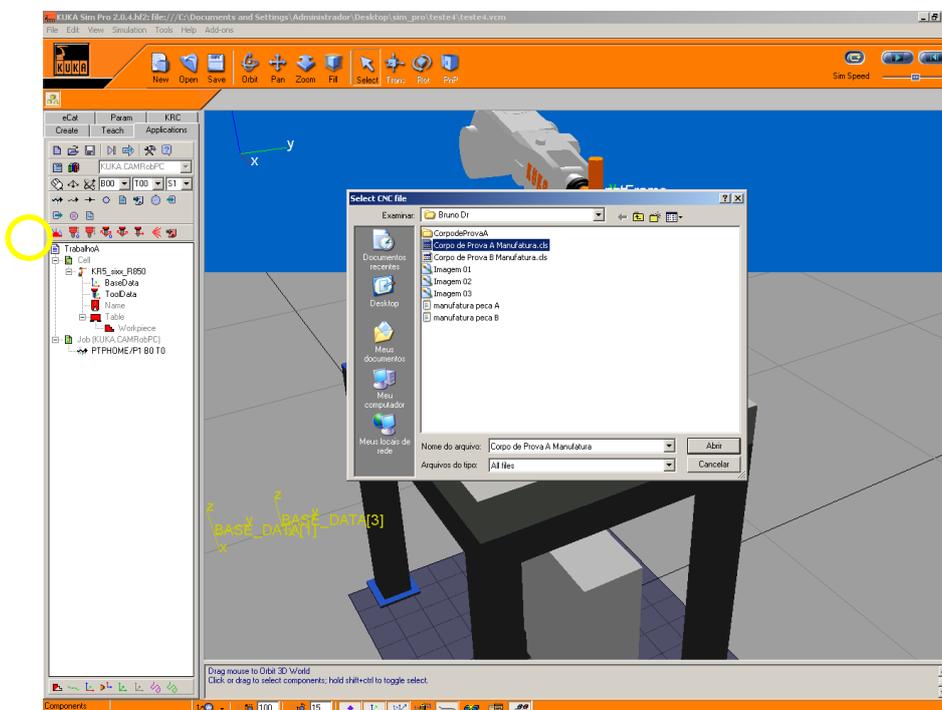


Figura 8.21: Add Process Comand

Na janela que se abre, deve-se definir o parâmetro de avanço rápido. Isso é, quando o sistema CAM envia avanço rápido, comando G0 para a maioria dos comando CNCs, o robô deve saber qual a velocidade que deve avançar. Isso porque o robô pode atingir diferentes velocidades máximas dependendo da programação. Se tiver que haver uma interpolação que exija muito movimento de um determinado eixo e pouco dos demais, a velocidade máxima será limitada pela velocidade máxima desse eixo nessa situação. Quando se faz movimento ponto a ponto, não importando a trajetória, o robô maximiza o movimento dos eixos de forma a que todos trabalhem de forma mais harmônica possível tirando proveito do máximo de cada eixo para atingir a posição da forma com a trajetória mais curta. Portanto, de forma a não estourar essa limitação e causar erro no comando, deve-se definir uma velocidade máxima. Para essa tese, foi definido 5000 mm/min. Os ensaios mostraram que o robô se comporta bem com esse avanço. A Figura 8.22 ilustra o campo alterado.

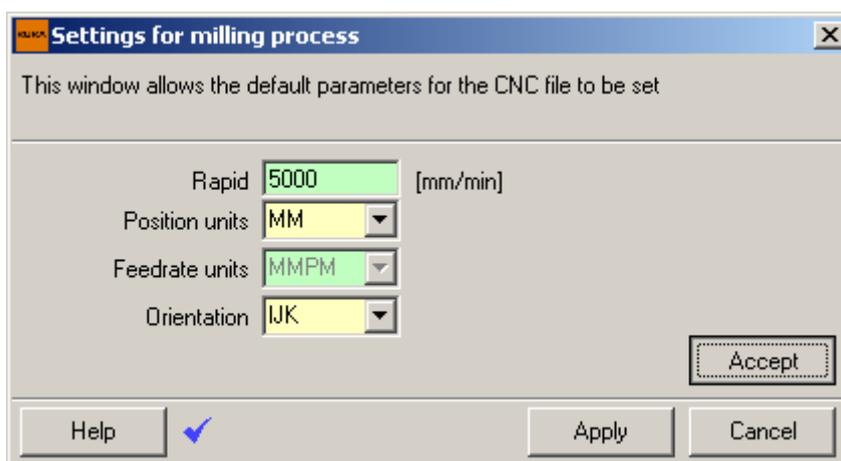


Figura 8.22: Parâmetros de velocidade em vazio

Ao selecionar **Accept** e **Apply**, aparece uma mensagem de erro. Se a mensagem não aparecer de imediato, na janela **CNC file processing**, selecionar **Read**. Algumas linhas do arquivo CLSF não são reconhecidas. Figura 8.23. Deve-se selecionar **Yes** para alterar o arquivo.

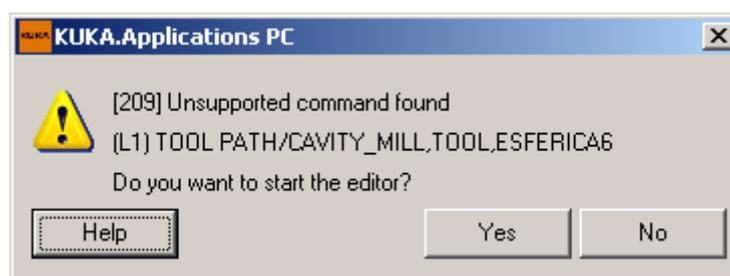


Figura 8.23: Erro no arquivo CLSF

Na janela que se abre, de editor em formato .txt. Deve-se apagar as linhas até o comando **paint/color, 6**. Conforme Figura 8.24.

```

Corpo de Prova A Manufatura.cls - Bloco de notas
Arquivo  Editar  Formatar  Exibir  Ajuda
TOOL_PATH/CAVITY_MILL,TOOL,ESFERICA6
TLDDATA/MILL,6.0000,3.0000,85.0000,0.0000,0.0000
MSYS/0.0000,0.0000,0.0000,1.0000000,0.0000000,0.0000000,0.
$$ centerline data
PAINT/PATH
PAINT/SPEED,10
LOAD/TOOL,2,ADJUST,1
PAINT/COLOR,186
RAPID
GOTO/-13.1990,30.0000,55.7362,0.0000000,0.0000000,1.0000000
PAINT/COLOR,211
RAPID
GOTO/-13.1990,30.0000,46.0695
PAINT/COLOR,6
FEDRAT/MMPM,15000.0000
GOTO/-13.1990,30.0000,43.0695
GOTO/-9.1990,30.0000,43.0695
PAINT/COLOR,31
GOTO/-9.1003,33.2686,43.0695
GOTO/-8.8047,36.5253,43.0695
GOTO/-8.3133,39.7582,43.0695
GOTO/-7.6278,42.9556,43.0695
GOTO/-6.7506,46.1058,43.0695
GOTO/-5.6852,49.1975,43.0695
GOTO/-4.4352,52.2192,43.0695
GOTO/-3.0052,55.1600,43.0695
GOTO/-1.4005,58.0093,43.0695
GOTO/0.3731,60.7566,43.0695
GOTO/2.4757,63.6037,43.0695
GOTO/4.7595,66.3075,43.0695
GOTO/7.2150,68.8564,43.0695
GOTO/9.8316,71.2397,43.0695
GOTO/12.5982,73.4471,43.0695
GOTO/15.5030,75.4692,43.0695
GOTO/18.5335,77.2975,43.0695
GOTO/21.6769,78.9240,43.0695
GOTO/24.9197,80.3419,43.0695
GOTO/28.2482,81.5452,43.0695
Ln 1, Col 1

```

Figura 8.24: Correção do arquivo CLSF

Na janela **CNC file processing**, a primeira etapa está em verde, significando que a leitura está correta do arquivo CLSF. A Figura 8.25 demonstra a leitura correta, e a mesma figura ainda ilustra através do círculo amarelo o próximo comando.

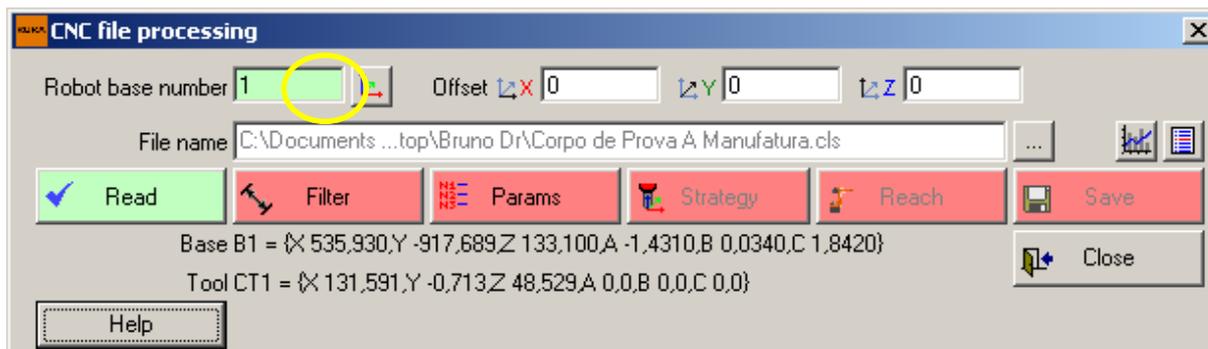


Figura 8.25: Comando Read finalizado.

No capítulo sobre zeramento de base do robô, essa, quando calibrada, zerada, gera uma tela com suas coordenadas perante as coordenadas globais do robô.

Essas coordenadas devem ser anotadas e copiadas na tela ilustrada pela Figura 8.26.

Esse passo é para efeito de simulação. Para o software simular o movimento do robô, ele deve saber onde é o zero peça. Então, deve-se entrar com esses dados no sistema. No caso dessa tese, foi escolhida a base 9. Essa escolha foi arbitrária com o argumento de escolha baseado em selecionar uma base mais distante na tabela das que já se tenham sido usadas por projetos anteriores.

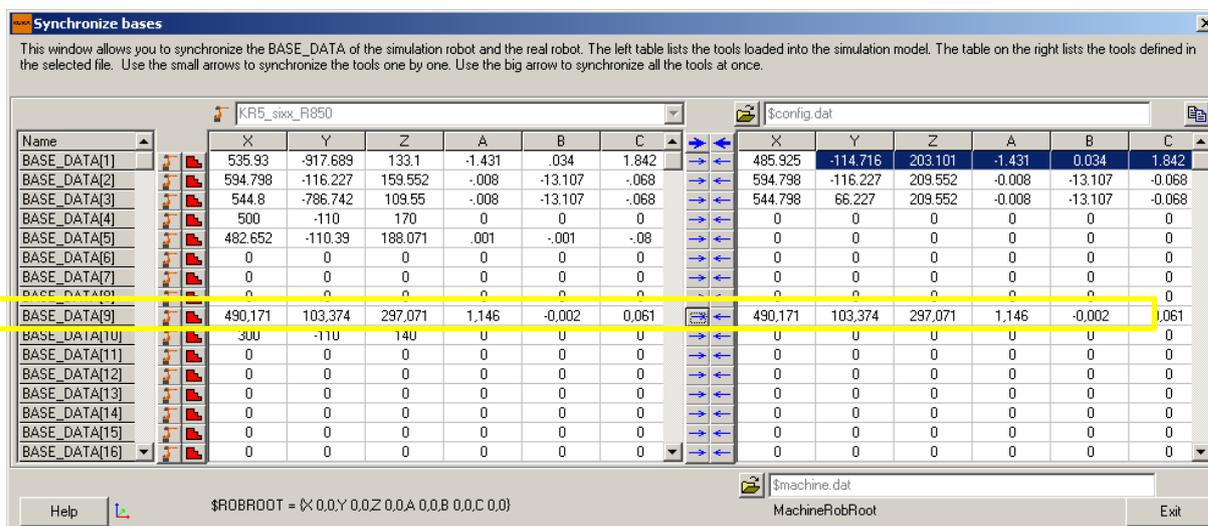


Figura 8.26: Janela de seleção e configuração de base

A Figura 8.27 ilustra a janela após a saída da planilha. Deve-se atentar a modificação da base para 9.

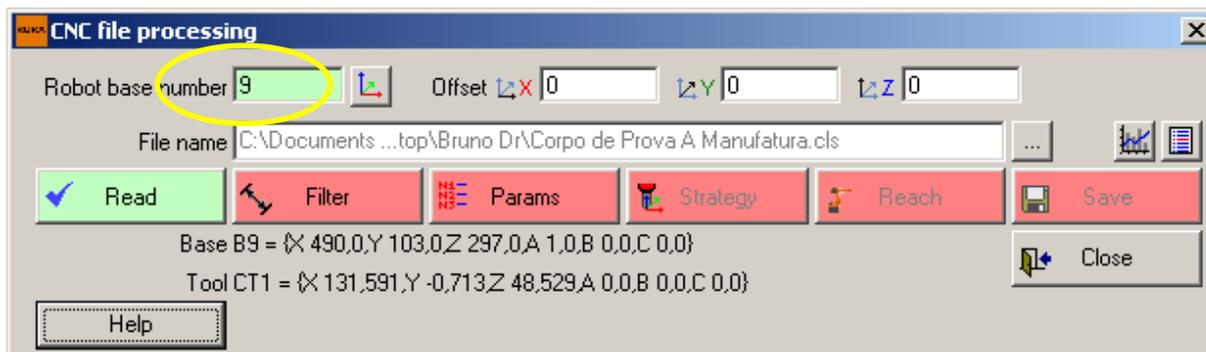


Figura 8.27: Estado depois da inserção das coordenadas da base

Ao selecionar **Filter**, foram escolhidas em ambas as partes da janela a opção Fine, e entrado com o parâmetro do CAM para a primeira janela de tolerância, 0,06. Esse valor pode ser diferente, mas foi colocado o mesmo como uma forma de referência. Conforme Figura 8.28.

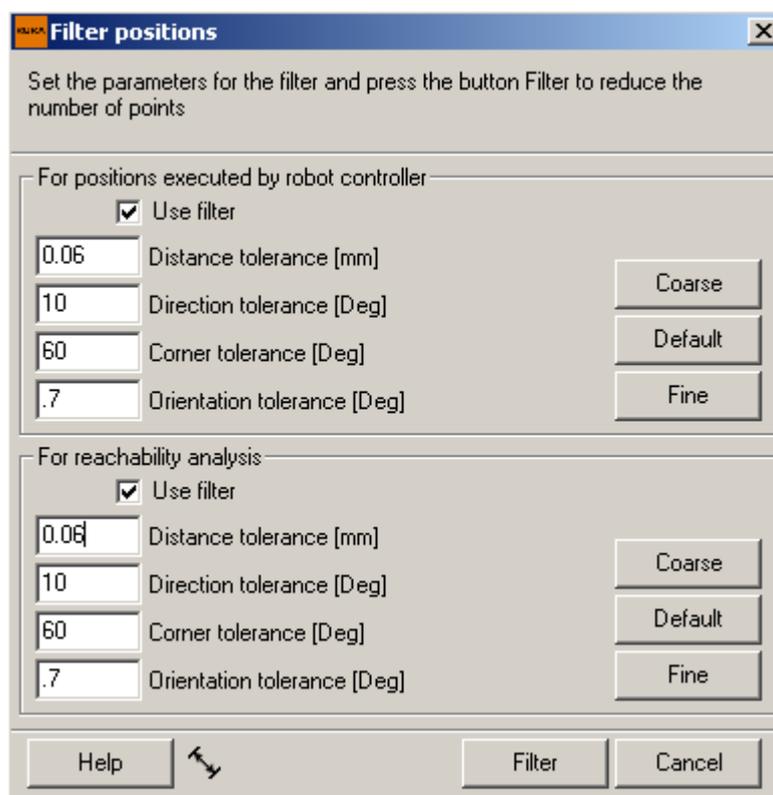


Figura 8.28: Janela de filtro

Após a configuração dessa janela, a janela inicial deve ficar conforme Figura 8.29.

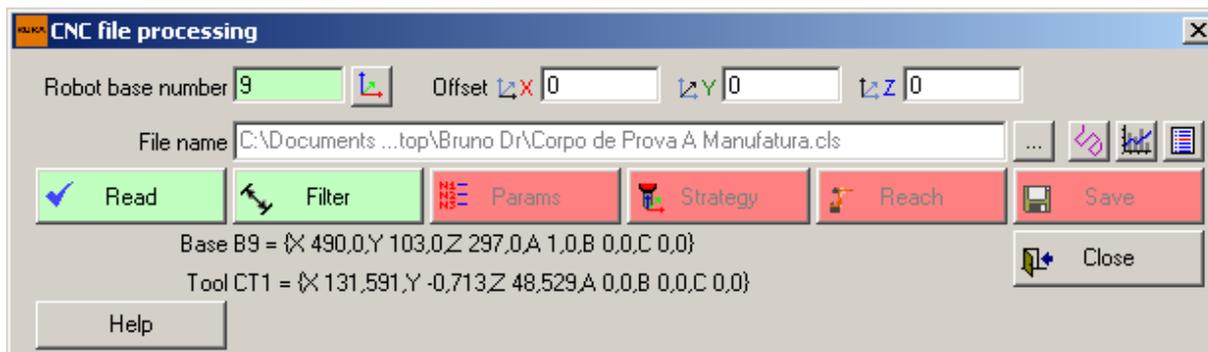


Figura 8.29: Final da janela Filter

A próxima configuração é a tela de **Params**. Nessa tela serão configurados as coordenadas da ferramenta, assim como a de base. E a rotação do *Spindle* para simulação. Nessa janela, conforme Figura 8.30, foi escolhido a ferramenta 11 e 15000 rpm.

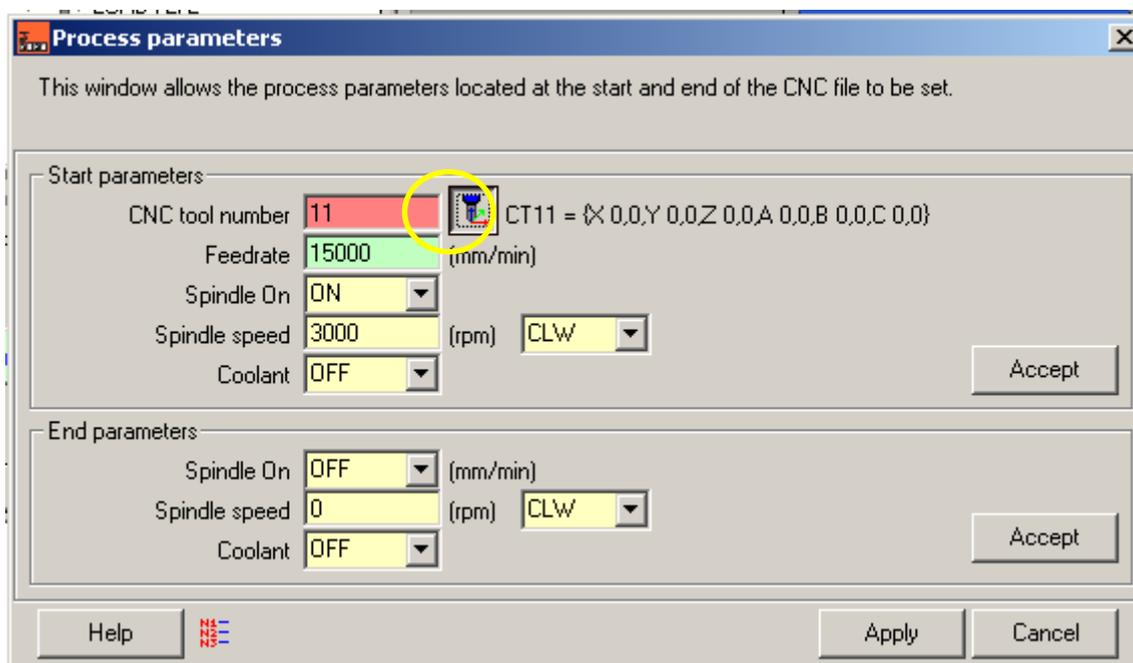


Figura 8.30: Janela de parâmetros de ferramenta

Após selecionar o comando na figura anterior circulado em amarelo, a janela ilustrada na Figura 8.31. Na ferramenta 11, da mesma forma que o procedimento da base, foram anotadas as coordenadas fornecidas pelo comando ao zerar a ferramenta e inseridas nessa janela.

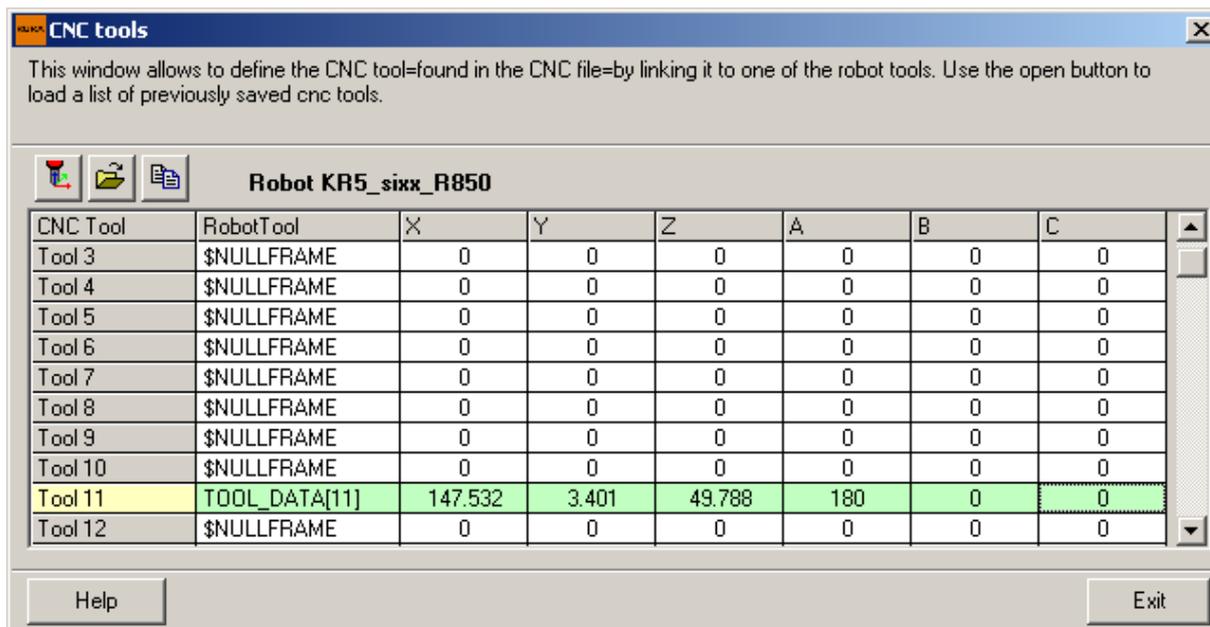


Figura 8.31: Parâmetros do zeroamento da ferramenta.

Após o preenchimento da tabela, a Figura 8.32 ilustra todos os campos da seleção em verdes.

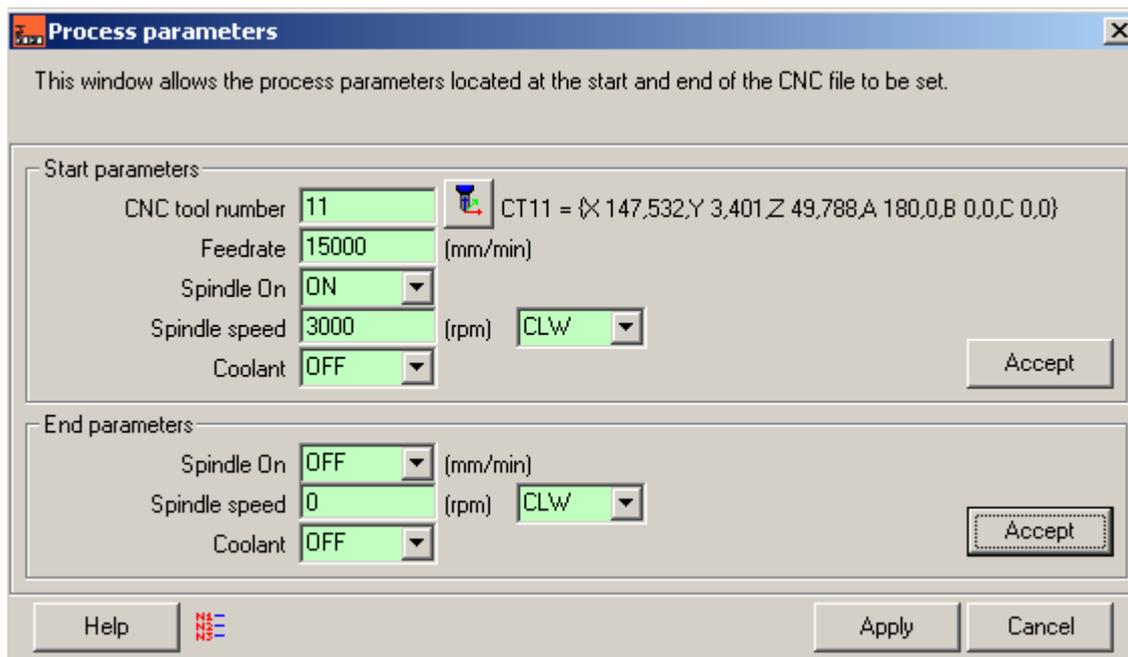


Figura 8.32: Finalização dos parâmetros da ferramenta

Agora, ao selecionar a janela Strategy, Figura 8.33, deve-se selecionar a opção YZ. Essa opção é a que restringe a questão do grau de liberdade a mais explicado no início desse tópico. Essa opção assim como o ângulo foi definido conforme a fabricação do dispositivo de fixação do *Spindle*.

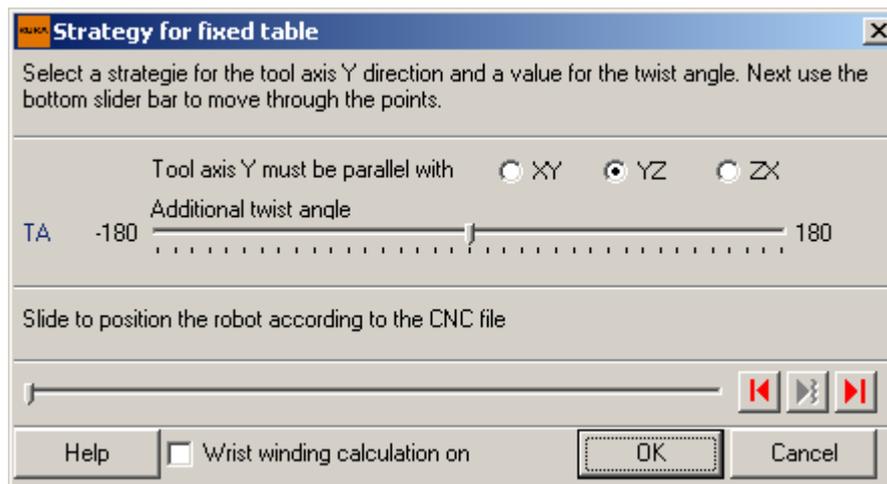


Figura 8.33: Janela de estratégias

A Figura 8.34 ilustra as diferenças de posicionamentos para a orientação do *Spindle*.

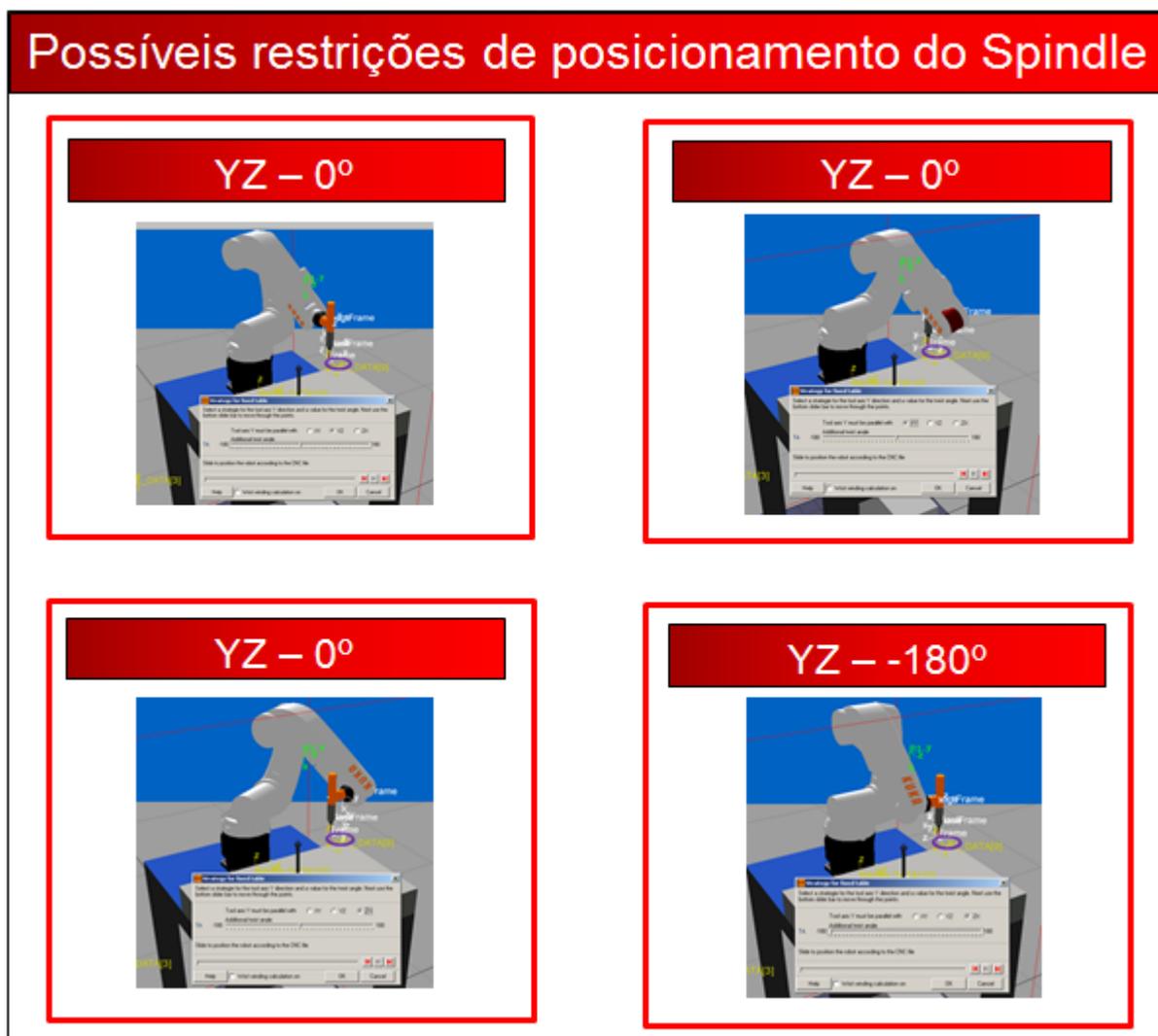


Figura 8.34: Possíveis restrições de posicionamento do Spindle

Dessa janela, se abre a janela ilustrada pela Figura 8.35 não há a necessidade de ser alterado nenhum campo.

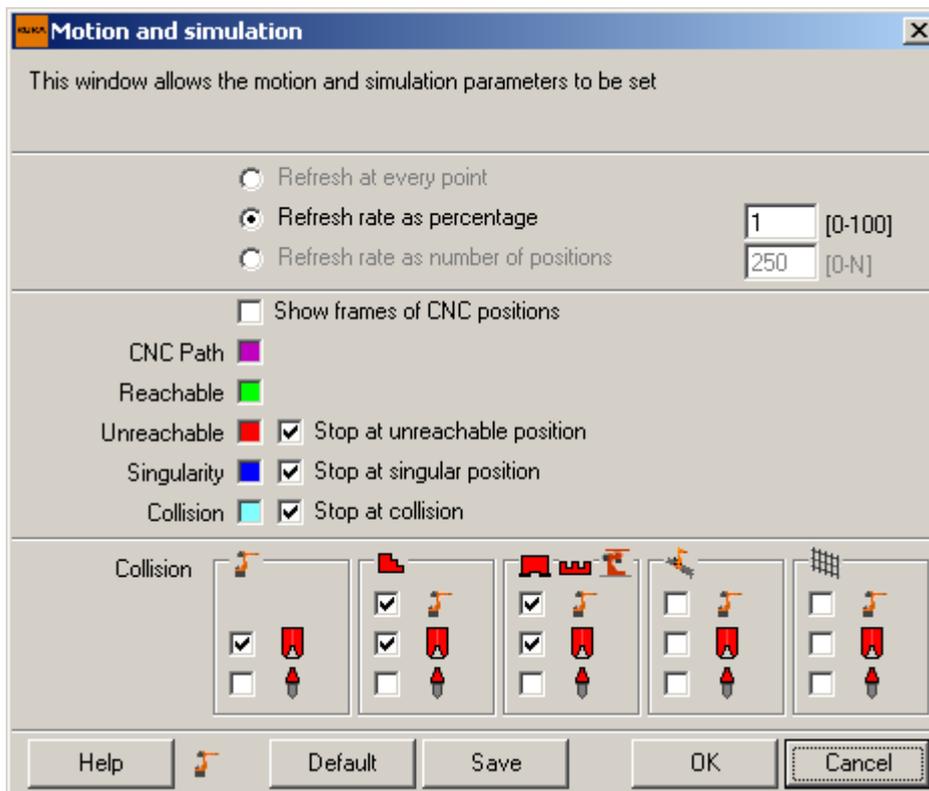


Figura 8.35: Janela de finalização das estratégias

Após todas as configurações feitas até então, a janela inicial deve ficar conforme a Figura 8.36.

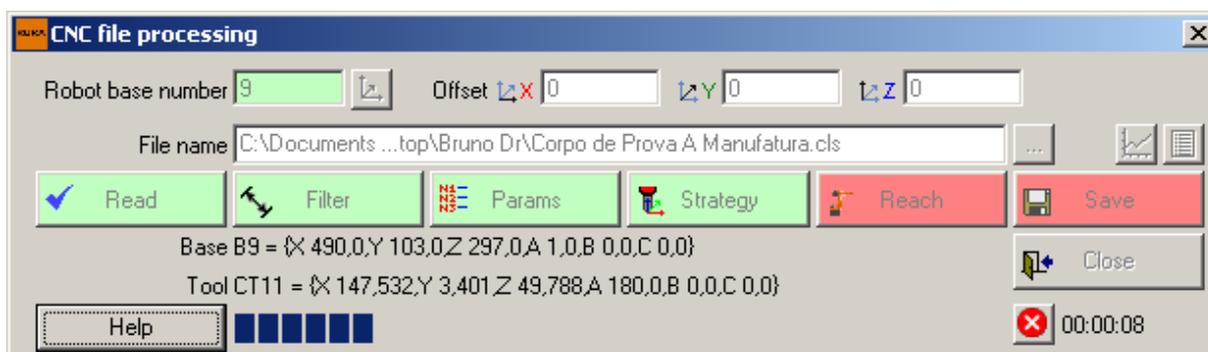


Figura 8.36: Estado da janela até a configuração da janela Strategy

Na janela Reach pode-se fazer a simulação da trajetória da ferramenta fornecida pelo sistema CAM. E se tudo estiver em ordem, conforme a Figura 8.37, basta salvar e fechar.

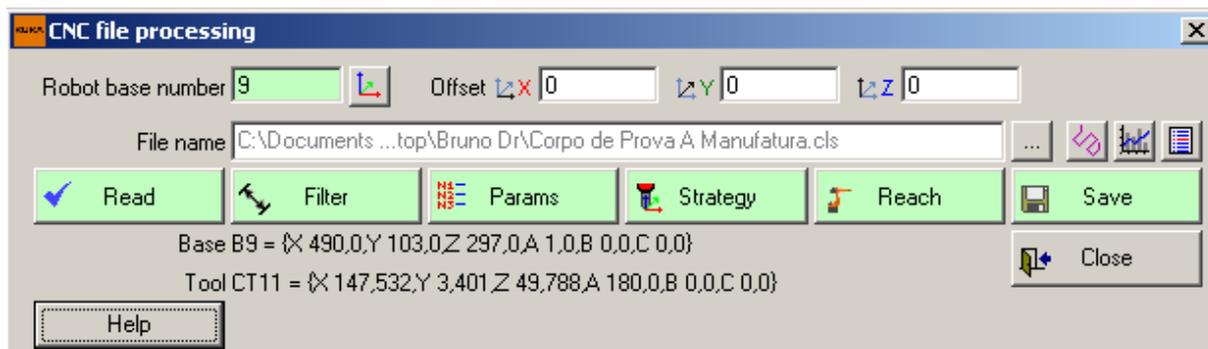
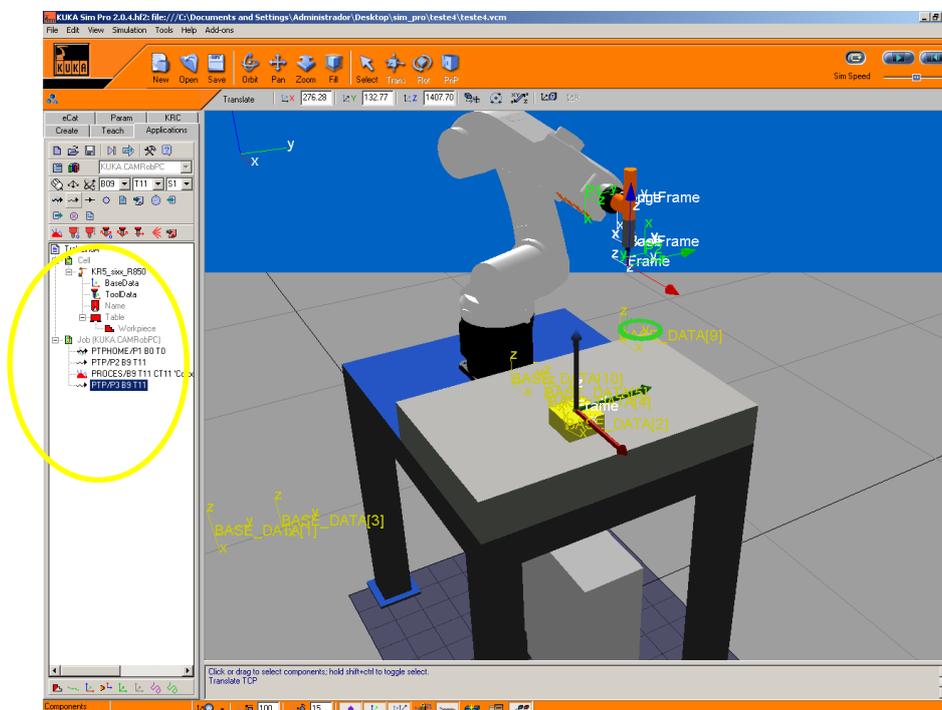


Figura 8.37: Processamento do arquivo CLSF em ordem

Após a realização dessas tarefas. É importante programar um ponto de início para o robô e um ponto de fim, conforme a Figura 8.37. Esse processo é mero para evitar colisões e ter ponto de início e fim conhecidos.



Através do comando **play**, e da janela que se abre, ilustrados pela Figura 8.38, pode-se simular todos os movimentos do robô.

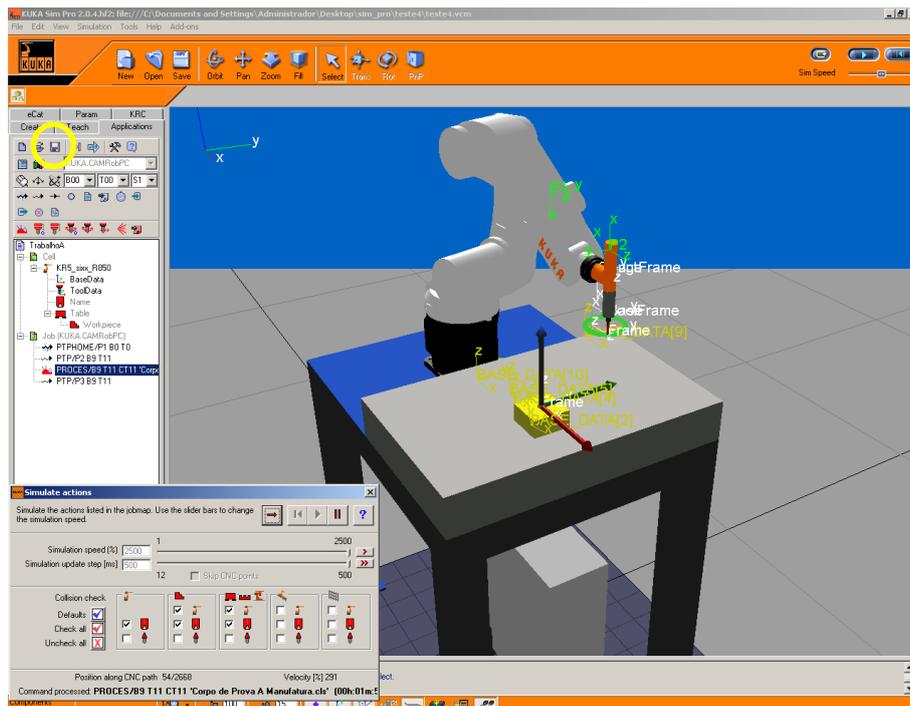


Figura 8.38: Simulação de todos os movimentos

Por último, através do comando indicado, pode-se finalmente salvar o arquivo pós processado no formato que o comando do robô reconhece.

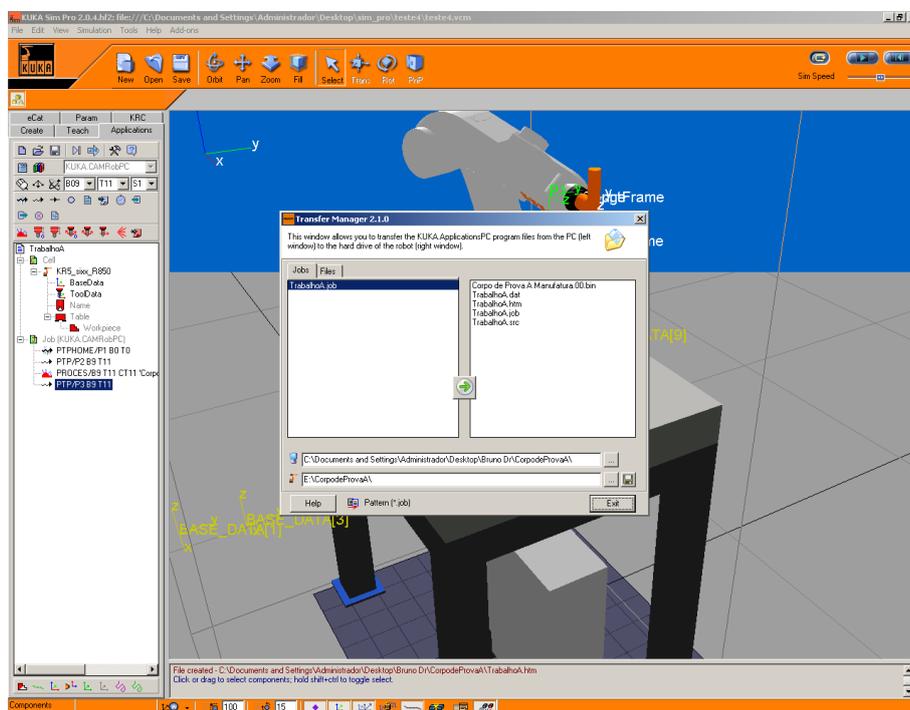


Figura 8.39: Geração dos arquivos para o robô